自由としてのフリー(2.0)

リチャード・ストールマンと自由ソフトウェア革命

著者　サム・ウィリアムズ/リチャード・ストールマン
翻訳　後藤洋 / 弓削光好

This is a Japanese translation of Free as in Freedom 2.0: Richard Stallman and the Free Software Revolution, a revision of Free as in Freedom: Richard Stallman's Crusade for Free Software.

目次

はしがき

リチャード・ストールマン

この版で私がしようとしたのは、私の知識をウィリアムズのインタビューや部外者の観点と結びつけることだった。これがどこまで達成できたかは、実際に読んで判断していただきたい。

私が本書の英語版を初めて読んだのは、2009 年に本書のフランス語版作成の手助けを頼まれたときだ。このとき、少なからぬ変更を必要とした。

多くの事実が訂正を必要とし、根本的な変更を要する箇所もあった。プログラマではないウィリアムズは、技術的、法律的に根本的に区別すべき点を曖昧にしていた。例えば、既存のプログラムコードを修正することと、そのアイディアの幾つかを新規のプログラムで実装することとの違いについてなどである。こ のため、旧版では、Gosmacs と GNU Emacs は、ともに PDP-10 Emacs を修正して作ったように言われているが、実際は、どちらも違う。また、Linux を「Minix からの派生物と誤った説明をしている。後日 SCO は IBM に対する悪名高い訴訟で、同じ誤った主張を行ったが、トーバルズもタネンバウムもそれに反論している。

第 1 版では、虚偽の感情を投影することで、多くのでき事を過度にドラマチックにしていた。例えば、私は 1992 年に Linux を「ひたすら敬遠していた」が、1993 年に Debian GNU/Linux のスポンサーとなることで、「劇的な方向転換」を果たしたとされている。1993 年に私が関心を持ったのも、1992 年に私が無関心だったのも、GNU システムの完成という同じ目的を追求するための実際的な（pragmatic）手段だった。1990 年に GNU Hurd のカーネルに着手したのも、同じ目的に向けての実際的な動機からだった。

こういうわけで、初版の記述には多くの誤りや矛盾があった。それらを訂正する必要があったが、全面的な書き直しをせずに整合性をもたせるのは、容易 なことではなかった。また、書き直さぬことが望ましくない理由は他にもあった。訂正を注解の形式で行うことも提案されたが、訂正量の大きい章が沢山あった ので、注解で行うことには無理があった。誤りによっては、注解で修正するには余りに広汎だったり根深いものだったりしていた。それ以外に、本文中に埋め込むものや脚注にするものは、本文を圧倒してしまう箇所があり、本文が読みにくくなった。脚注を追うのにうんざりした読者は読み飛ばすだろう。そこで、私は 直接本文を訂正することにした。

とはいえ、私の知識の埒外の事実や引用まで全て点検しようとしたわけではない。こうしたもののほとんどは、あっさりウィリアムズの権威に委ねている。

ウィリアムズの旧版には、私に批判的な引用が沢山含まれていた。私はこれら全てをそのままにしたが、適宜反論を加えた。私は引用の削除は行なっていないが、第 11 章では例外的に、オープンソースに関するもので、私の人生や仕事と関係のない箇所を幾つか削除した。

同様に、ウィリアムズ自身の解釈で私に 批判的なものも、事実や技術の誤解にもとづくものでない限り、そのままにした（コメントを加えたものはある）が、私の仕事や思想や感情に関する不正確な言 明は正した。彼が個人的な印象として提示したものはそのままにしており、この版の本文で「私」は常にウィリアムズのことであるが、そうでない場合は、「RMS:」と注記している。

この版では、GNU と Linux を組み合わせたシステム全体を常に「GNU/Linux」と表記し、トーバルズのカーネルをいう場合は常に「Linux」とした。ただし、引用中で他の表 記に「原文のまま」と注記した場合を除く。システム全体を「Linux」と呼ぶのが、なぜ誤りかつ不公正であるかの説明は、http://www.gnu.org/gnu/gnu-linux-faq.html を見てほしい。

ジョン・サリバンが有益な批判と助言を沢山してくれたことに感謝する。

序文

サム・ウィリアムズ

今夏で、『自由としてのフリー』という書物の始まりとなった電子メールの交換から十年になる。それは『リチャード・ストールマンと自由ソフトウェア 十字軍』になり、その拡大版の、ここに序文をしたためている作品の『リチャード・ストールマンと自由ソフトウェア革命』となった。

いうまでもなく、その十年間に多くの変化があった。

最初に、アメリカ至上主義の時代に誕生した本書の筋立ては、一人の男の予言者エレミアのような奮闘を描いている。それがほとんど懐かしく思えて、少 数の周辺的懐疑論者の批判を封じ込め、テクノ資本主義システムが素晴らしく機能しているかに見えた当時への帰還のように感じられる。彼は経済的観点からで はないかもしれないが、倫理的観点から、商業システムの短慮について仲間のソフトウェア開発者を啓蒙していた。それは自由の領域にあって計算機科学を誕生 させた知的文化を、占有的で閉鎖的なコミュニティの粗野な凝集物へと変貌させることにきゅうきゆうとしているのだった。

今では、そのシステムを疑うことはほとんど当然の美徳になり、それは、この十年間に一貫していたものがあるとすれば、その物語の糸を見るのに役立つ。

以前ほど密着してソフトウェア産業を追っていない私だが、当時の状態にも増して、今日では目が釘付けになるようなことが一つある。それはクールなテ クノロジー「プラットフォーム」や最新のネットワーク・トレンドに乗るのと引き替えに、広大なプライバシー関連情報とユーザーの個人の自由を進んで引き渡すことに示された一般消費者の気楽さだ。

数年前なら、これをアップルの共同創業者スティーブ・ジョブスの無敵の成功に敬意を表して「iPod 効果」と呼んだかもしれない。 ジョブスは、音楽業界とデジタル音楽リスナーとの間にあった積年の相互不信と敵対関係を脇に追いやり、音楽業界とデジタル音楽リスナーをアップルの iPod という単一の魅力的な装置と iTunes という制限的なライセンス制度の回りに呼び集めた。 今、雑誌や新聞に記事を書くなら、ブランド名の進化に遅れず、新聞ジャーナリズムや電子書籍出版の領域で起こっている並行的で地殻変動的な変化を認知するためにも、 たぶんそれを「iPad 効果」とか「Kindle 効果」と呼ぶ必要があろう。

上述のブランド名を無闇に宣伝していると受け取られないように、RMS は、特にソフトウェアの自由の領域でそれらがもつ多くの欠点をていねいに説明している二つのウェブサイトに等しく言及することを提案する。私は、この均等 な時間配分の精神に賛成した。彼が推奨するウェブサイトは、DefectiveByDesign?.org と BadVista?.org だ。

根拠の有無にかかわらず、企業ブランドは急速に変化する世界のソフトウェア品質の唯一の保証人だとする観念は、大半の企業ブランドが歴史的最安値で売却されている時代でさえ、依然として抜き難いものの一つである。

十年前、技術カンファレンスで、ストールマンを含む古顔の提供する別の選択肢の可能性

の説得的ヴィジョンの対話（または直接的な指導）に人々が耳を 傾けているところをよく見かけた。ようやく気付いたが、我々ジャーナリズム・ゲームの新米が使い方をやっと覚え、鼻にかけていたツール、マイクロソフト・ ワード、パワーポイント、インターネットエクスプローラは、これはよく引用される単独のソフトウェアベンダーの人気製品をいくつか列挙したものにすぎない が、パーソナル・コンピュータのオリジナルの設計者たちが着手したそびえ立つ大建造物の淡い影に過ぎない、と我々に認識させたのはこれら古顔の仕事だった。

今日、目前には、ほとんど正反対の状況があるように見える。大建造物は今や四方に広がる一つの生態系だ。 そこはジャングルで、アイデアは満ち溢れていても、 持続可能な成長のための安定した隙間をほんのわずかしか提供しない。景観からいつまでも消えない構造的欠陥、アップルによるアイフォーンやアップストアの 独裁的監視、グーグルの「邪悪」の定義の変更に対して不満を語る多くのハッカーは今も存在している。しかし、このホッブズ的王国には、企業のガイダンスを 喜んで信じる「生まれたときからデジタルな」消費者という新鮮な収穫物が毎年運ばれて来る。もしかすると、かつてデスクトップ・コンピュータを使用中に歯ぎしりさせられたような問題が、大方、自由ソフトウェアの助けで解消されたからなのかもしれない。

ともあれ、消費者のソフトウェアの信頼性が改善されるにつれ、消費者の嗜好の一歩先に留まるレースは、アプリケーション開発者を金銭的利害にますま す縛りつけるようになった。私はハッカー精神がもはや存在しないとか、少しでも弱まったと言っているのではない。ただ、キーワードがスポンサーのページを 指すように "info" ページを書き換え、起動時に八割の率で意味エラーを引き起こすようなフェイスブックのアルゴリズムを作ったプログラマは、新車のポルシェで多くの時間を費 やしながら、「スーツ」が邪魔しなければプログラムが本当は達成しえたはずのことについて不満を鳴らしているのではないか、と言っているのだ。

今や数百万の人々が自由ソフトウェアのみが走っているコンピュータで自由ソフトウェアを実行しているのは真実である。しかし、一般消費者の視界から は、「ソフトウェア」や「コンピュータ」という用語は、ますます遠いものになってきた。2010 年の時代の携帯電話の多くは、同じ金額で機能面において 2000 年の時代のラップトップに匹敵する。それなのに、携帯電話を購入するとき、どれほどのユーザーがコンピュータやソフトウェア、オペレーティングシ ステムが可能にする機能性に思いを馳せるだろうか。現代の電話ユーザーの広大な多数派は、提供されるアプリケーションの数、ネットワークの堅牢性、そして 最も重要な月々の料金プランをもとに購入を決めている。

この状況で、個人の便利さに反しても、個人の自由というレンズを通してソフトウェアの購入を検討する消費者を獲得することは、さらに困難になったのではないとしても、確実に、より複雑な試みになっている。

この悲観的調子の序文を与えられて、人はどうしてこの先を読みたいと思い、この本を読

むべきだろうか。
私は、二つの大きな理由を提供できる。
最初の理由は、個人的だ。『自由としてのフリー』のエピローグに記したように、あの本の出版の少し前に友情的とはいえない形でリチャードと私の連絡 が途絶した。責任の大部分は私にあった。伝記的スケッチが自由ソフトウェアの諸原理と衝突しないようにリチャードと一緒に作業をしていて、-- 著作権の仕組みとして GFDL を使った最初の著作の一つとして『自由としてのフリー』を作ったのは成果の一つだったと誇りを持って言いたいが -- 本を編集して、リチャードから届いた誤り訂正の長大なリストと明確化の注文を取り込む段になって、突然、私は協力関係を終わらせてしまった。
著作者の独立性やジャーナリスティックな客観性の原理の水面下に身を隠すこともできたが、出版社のオライリーに時間の猶予をもらわなかった事を私は ずっと後悔していた。オライリーはすでに私の主な契約条件の一つ（GFDL）を承諾して、時間ぎりぎりの大量の変更に耐えており、私は自分の運をさらに試 すことをためらった。
『自由としてのフリー』の出版から間もない数年間は GFDL を引合いに出して自分の決定を正当化できた。GFDL はちょうどソフトウェアの領域の GPL のように、いかなる読者にも、本を修正し、それを競合作品として販売することを可能にしている。かつてアーネスト・ヘミングウェイが言ったように、「最初 の原稿はどれも糞だ。」ストールマンやハッカー・コミュニティの誰かが『自由としてのフリー』をせいぜい初期原稿と見てくれれば、そう、少なくとも、私は 彼ら自身が最初の原稿を書く時間と手間を節約させたことになる。
今、実にそのリチャードが大規模な書き直しをしたものを届けてくれた。当時の自分は変えようがなく、私は彼の努力を賛嘆するばかりだ。心からお礼を 申し上げる。あと一つだけ残る希望としては、リチャードの仕事のスピードが落ちる兆しはないので、この合作にさらに考証が付加されていくことを期待した い。
二番目の理由に移る前に、この本のうれしい副産物を記しておくべきだろう。リチャードと私の間にメールによるコミュニケーション・チャネルが再開された。そのおかげで私はストールマンの鋭利な叙述スタイルを再認識した。
例証的な逸話で、本文の中でリチャードと論争していた人々なら誰でもおそらく楽しめる逸話、マウイ島のキヘイの交通渋滞でリチャードが冷静な態度を 失っていくプロセスを見聞して記録した最初の本の第七章（『ハッカー地獄巡りの小旅行』）になった文章を議論するうちに、私は本書の評者たちが共通に漏らすある指摘に気付いた。それは、場違いなエピソードに見えたとか、一冊の伝記を中断する断片的な雑誌風の横顔紹介だというものだった。リチャードにその理 由だけでもエピソードを捨てられると言ったのだが、それでも入れたのには正当な理由が二つあったことを付け加えた。一つは、警告されていたものの自分で直 接体験したことのなかった、ストールマンの気質を垣間見せてくれること。二つ目は、シーンの全体が隠喩的な価値があると感じたことだ。あの章のタイトルは そのためだ。
ストールマンは、驚いたことに、どちらの理由も認めてくれた。それより彼がひっかかっ

たのは、二つのささいな言葉だった。私はある個所で、我々の二 台の車のキャラバンを先導するドライバーが「わざと」袋小路に連れ込んだと非難している彼の言葉を引用した。それが本当なら、その非難は実情以上の敵意の レベルを感じさせるものだ。あのときは紙ノートしか持っていなかったので正確な記録に頼ることができないが、ストールマンの実際の言葉を取り違え、本来の 意味よりひどいものにしていたかも知れない。

他方、それとは別のことで、ストールマンは彼の発言として "fucking" が使われていることに疑問を投げかけてきた。このときも、テープは持っていなかった。しかし、私は、リチャードがニューヨーク出身であること思い出させ る、印象的な冒涜的態度をはっきりと思い出すし、私の記憶を支持するつもりだと返事した。

翌日、リチャードからメールの返信があり、批判がまた述べてあるので、私は最初のメッセージを読み返さざるをえなかった。それで判ったのは、ストールマンは "fuck" のところを問題にしているのではなく、引用の "-ing" の部分を問題にしているのだった。

「[その言葉を]疑う理由の一つは、"fucking" を副詞として使っていることに関わっています」と、ストールマンは書いていた。「私は決してそのように話しません。それでその言葉は何か別のものだったと確信しているのです。」

まいった。

この本を読まぬわけにはいかないと感じるべき二番目の理由は、この序文の最初の主題に戻る。2010 年に我々の前にある未来は、今も振り返ると目に 浮かぶ 2000 年の未来と何と違っていることだろう。私は正直でありたい。多くのアメリカ人（そして非アメリカ人）のように、私の世界観は、2001 年 9 月 11 日の事件によって変わった。それはとても大きかったので、『自由としてのフリー』の出版後間もなく、私の関心は、自由ソフトウェア運動とそれを正しい方向へ進ませようとするストールマンの努力とは全く別のことへ移っていった。私は大きな傾向と主な論点はなんとか追い続けていたが、ソフトウェア標準を めぐる日々のドラマやソフトウェア著作権やソフトウェア特許は、大部分を読み飛ばしてしまう何かになっていた。要するに、インターネットのニュースに対する関心は地方紙の水道委員会の記事並みになっていた。

[RMS: この本ではこれ以降 2001 年 9 月の攻撃は言及されないが、ここに短くコメントしておく価値はある。多くの言明のように「すべてを変化させる」ものとはほど遠く、攻撃は実際のところアメリカをわずかに変えただけだった。権力の座には相変わらず我々の自由を嫌うならず者たちがいる。主な違いは、今は、彼らが 自由を奪う法律の口実として「テロリスト」を持ち出せるようになったことにすぎない。これについて、さらに読みたい方は、stallman.org の政治ノート（the political notes）を参照してほしい。]

これは大部分嘆かわしい展開だ。なぜなら、10 年かかってやっと透明であると信じられる光の中で、私は成熟しつつある 21 世紀を見ているからだ。あらためて、編集者に売り込む言葉にするなら、私はそれを『プロセスの世紀』と呼ぶだろう。

これによって、私たちは歴史上の稀な地点に立っていると言いたい。そこでは、冷笑主義

をすっかり離れてみると、世界を本当に変える力は普通の市民レ ベルに委ねられている。キャッチはもちろん、あなたに属している同じ力は他の誰にでも属しているということだ。過去の時代には、人は少数の適所にいるイン サイダーの共感を手に入れれば良かったが、今日の改革者は、競合するアイデアと利害のベクトル場全体を調整しなければならない。要するに、今日、効果的な 改革者であるためには、十年またはそれ以上に渡ってタイタンのようなスタミナで荒野で叫び続ける意欲以上のものが必要であり、長持ちのする伸縮自在なアイ デアを理路整然と語る方法と、システムをシステム自身のゲームで打ち負かす方法を知っている必要がある。

お手本ではないかもしれないが、最も控えめに言っても、あらゆる点でリチャード・ストールマンは、まさしくその成功する改革者の原型であると言いたい。

どんな問題も正されるまでに、委員会の会議と小委員会での聴聞に何十年もかかるような未来を嘆く人もいるかもしれない。しかし、たとえば私には、別の未来も見える。 それは、個人や小グループの行動にとても敏感に反応するので、 ついに自ら任命したアクターがその反応を試してやろうとするような未来だ。 考えただけでもぞっとする。

要するに、あなたが私のように、21 世紀が 20 世紀よりも流血の少ない道を歩むのを見たいと願うタイプの人間なら、多くのじれったい外観を見せてはいるものの、 水道委員会こそ、現在その戦いが行われている場所である。 本書第一章冒頭のヴェルギリウスの題辞も示唆しているが、この本はインターネットの時代の叙事詩になるかもしれないという期待を私はずっと持っていた。それは、英雄的だが欠点もある中心人物をめぐって繰り広げられ、時とともに作者の刻印は霞んでいくような叙事詩である。

そこで、私は、この序文を終えるときのいつものやり方、つまり、本文をさらに良くしたいと望むすべての読者からの貢献と変更を注文することでこの序 文を終えたい。付録 B の GNU 自由文書ライセンス（GFDL; GNU Free Documentation License）があなたの権利や、変更を提案し、訂正を加え、さらには別の独立ヴァージョンをつくることの案内をしてくれる。もしも、リチャードまたは 私を通じて変更することを望むだけなら、自由ソフトウェア財団のウェブサイトに適切な連絡情報がある。

それではみなさん、グッド・ラック。この本を楽しまれよ！

サム・ウィリアムズ

スタテン島、アメリカ合衆国

ギリシア人はこわい、贈り物を持ってきても。
ヴェルギリウス『アイネーイス』

第一章　プリンタがなくて

新しいプリンタが、また紙づまりを起こした。

マサチューセッツ工科大学人工知能研究所（ AI ラボ）の職員プログラマ リチャード・M・ストールマンは、その故障を発見した。こいつは辛い。27 歳のストールマンは、オフィスのレーザープリンタに 50 ページのファイルを送信した 1 時間後に、生産的な作業セッションを中断してドキュメントを取りに行った。着いてみると、プリンタのトレイには 4 ページしか出ていなかった。しかもその 4 ページは他のユーザーのものだったので余計がっかりしたが、それはストールマンの印刷ジョブと他の誰かのジョブの残りの部分がラボのコンピュータ・ネット ワークの電子配管のどこかに詰まったままになっているということだった。

ソフトウェア・プログラマが機械に待たされるのは仕事上の災厄なので、ストールマンはそれほど気にしない。しかし、機械の処理を待つのと、機械の世 話をするのではかなり違う。プリンタの脇に立って印刷されたページが一枚ずつ出てくるのを見つめているのはこれが初めてではなかった。日夜機械とそれを制 御するソフトウェアの性能改善に没頭している人間として、ストールマンは機械を開けて中身を見ることで、この問題の根本原因をつきとめたい、という自然な 衝動にかられた。

残念ながらコンピュータ・プログラマとしてのストールマンの力量は、機械工学の領域にまでは及んでいなかった。新たに印刷された文書が機械から流れ出てくる間、ストールマンは、このプリンタの紙づまり問題を回避する別の方法を考えてみることにした。

AI ラボのスタッフたちがこの新しいプリンタを両手を広げて歓迎したのはどれぐらい前のことだっただろう、とストールマンは考えてみた。この機械はゼロックス 社から寄贈されたものだ。高速のゼロックスコピー機を改造したもので、最先端の試作品だった。コピーをするのではなく、コンピュータ・ネットワークを通して送られて来るデータをプロの仕上がりの文書に印刷してくれる。世界的に有名なゼロックス・パロアルト研究所のエンジニアたちが開発したもので、その後の 10 年間でコンピュータ産業をとらえたデスクトップ印刷革命の最初の味がした。

最新最高の装置で遊びたいという本能的欲求に突き動かされて、 AI ラボのプログラマたちはこの新しい機械を直ちに、 研究所の洗練されたコンピュータインフラに組み込んだ。結果は大満足だった。研究所の古いレーザープリンタとは違い、新しいゼロックスの機械は高速だった。毎秒 1 ページの割合で飛び出し、20 分かかった印刷は 2 分になった。新しい機械は精度も良かった。円は楕円ではなく、円に見えた。直線は小振幅の正弦波ではなく、直線に見えた。

そのプリンタは、どこをとっても、抵抗し難い魅力的な寄贈品だった。しかし機械が使われ始めると、欠陥も露呈してきた。大きな欠点は、その機械が紙づまりを起こしやすいことだった。エンジニアリングに関心のあるプログラマたちは、その欠陥の理由をすぐに理解した。コピー機ならふつう機械を直接監視する操作者が要る。紙づまりが起こっても、それを直してくれる操作者がいつも近くにいると考えたゼロックスのエンジニアたちは、自分たちの時間と精力を他の 難問の解決に捧げることにしたのだ。工学用語でいうと、ユーザーの労力がシステムに組み込まれていたわけだ。

コピー機をプリンタに改造するにあたってゼロックスのエンジニアたちは、ユーザと機械の関係を微妙に、だが本質的に変えた。機械が個々の操作者に仕 えるのではなく、ネットワーク上の操作者全員に仕えるようにした。機械のすぐそばに立っているのではなく、ネットワークの末端にいるユーザは、バケツリレー式につながる機械を通して印刷コマンドを送信し、欲しい印刷内容が目的地に適切な形でたどり着くことを期待した。実際に印刷されたのがほんの一部にすぎないとしても、最終的に出力結果を確認しに行くまで分らなかった。

AI ラボの住人でその問題に気づいたのはストールマンだけではなかったが、彼は問題の解決策も考えた。数年前、ストールマンはラボの一つ前のプリンタのため に、小型の PDP-11 マシンのプリンタ制御ソフトウェアと、メイン・コンピュータの PDP-10 で動く非互換時分割システムに修正を加えることで、同様な問題を解決したことがあった。ストールマンは紙づまりは解決できなかったが、PDP-11 に定期的にプリンタをチェックし、紙づまりを起こしていたら PDP-10 に報告させるソフトウェア・コードを挿し込むことができた。また、ストールマンは PDP-10 にコードを挿入して、印刷ジョブが待ちになっているユーザ全員に、プリンタが紙づまりを起こしていることを通知するようにした。通知の内容は「プリンタが 紙づまりを起こしています。直してください、」といった簡単なものだったが、その問題を解決する必要に最も迫られている人々に知らせが届いたので、すぐに が誰かが直してくれるチャンスがあった。

改良を進めるときのストールマンの方法は、直線的ではないがエレガントだった。それは問題の機械的側面は直さないが、ユーザーと機械の間に情報の経 路を設ける、という次善策だった。ソフトウェア・コードが数行追加されたおかげで、AI ラボの職員は、プリンタをチェックするために小走りに行き来することで毎週失っていた 10 分から 15 分の時間を節約することができた。プログラミング用語でいえば、ストールマンの方法はネットワーク全体にわたって拡張された知性を利用するものだった。

「そのメッセージを受け取ったら、誰か他の人が直してくれる、なんて言っていられないでしょう。」このロジックを回想してストールマンは言う。「自 分がプリンタのところに行かなきゃならないんです。プリンタにトラブルが起こって一、二分もすると、メッセージを受け取った人達が二、三人、機械を直しに やって来ます。この二、三人の内、少なくとも一人は問題の解決方法を知っているものです。」

こういう賢い解決法は、AI ラボとその土着民たるプログラマたちのトレードマークだった。実は、AI ラボの最も優れたプログラマたちは、「プログラマ」という言葉を軽蔑して、よりくだけた職業名であるハッカーを好んでいた。この肩書きは、創造的な悪ふざけから、既存のソフトウェアやコンピュータ・システムの改良に至る、あらゆる活動をカバーしていた。だが、この肩書には、ヤンキー的創意工夫という古風な 観念が暗に含まれていた。ハッカーにとって、動作するソフトウェアプログラムを書くことはほんの始まりに過ぎない。ハッカーは、それ以上のことに挑み、プ ログラムをとんでもなく速く、小さく、強力で、エレガントなものにするとか、賢い方法という点で何か印象的なものによって、自分の賢こさを見せつけ（そして他のハッカーに印象づけ）ようとする。1

ゼロックスのような企業はその製品（とソフトウェア）を、ハッカーが典型的に集まる場所に寄付することを方針にしていた。ハッカーがこれらの製品を 使うと、あとでその企業のためになるような仕事をするかもしれなかった。60 年代や 70 年代の初めには、ハッカーたちはメーカーが他の顧客たちにも配布で きるような、有益なプログラムを開発することがあった。

ゼロックスのレーザープリンタに紙づまりを起こす傾向があることに気づいたストールマンは、このプリンタに旧来の修繕方法を適用する、つまり「ハッ ク」しようと思った。だが、ゼロックス・レーザープリンタのソフトウエアを調べるうちに、厄介なことを見つけた。プリンタには、少なくともストールマンや 同僚のプログラマが読めるようなソフトウェアがなかったのだ。それまではどの会社も、礼儀として、ソースコード・ファイルを公開していた。それは可読なテ キストファイルで、機械が何をすべきか伝える個々のソフトウェア・コマンドを文書化していた。だが今度、ゼロックスはソフトウェアをコンパイル済み、つま りバイナリ形式のファイルで提供していた。プログラマがファイルをのぞいても、延々と続く 1 と 0 の羅列しか見えず、ちんぷんかんぷんだったろう。

0 と 1 の羅列を低水準の機械語命令に変換する「逆アセンブラ」と呼ばれるプログラムがある。しかし、これらの命令が実際に何を「する」のかを思い浮 かべるのは、「リバースエンジニアリング」として知られる時間のかかる難しい仕事だ。このプログラムをリバースエンジニアするには、つまったプリンタの出 力 5 年分の価値以上の時間がかかるかもしれなかった。ストールマンはそこまで熱心にはなれなかったので、その問題は一旦棚上げにした。

ゼロックスの非友好的な方針は、ハッカーコミュニティの通常のやり方とはひどく対照的だった。たとえば、古いプリンタを走らせていた PDP-11 用のプログラムや、表示端末を制御する別の PDP-11 用のプログラムを開発するために、AI ラボは、メイン・コンピュータ PDP-10 上に PDP-11 用のプログラムを作成するためのクロスアセンブラプログラムを必要としていた。それはラボのハッカーにも書けただろうが、ハーヴァードの学生だったストー ルマンは、そうしたプログラムをハーヴァードのコンピュータ研究室で見つけた。そのプログラムは同種のコンピュータ PDP-10 で走らせるために書かれていたが、オペレーティング・システムは異なっていた。ソースコードに書いていなかったので、誰がそのプ

ログラムを書いたのかは知るよしもなかった。だがストールマンはそのコピーを AI ラボに持ち帰り、AI ラボの非互換時分割システムでも動くようにソースコードを書き変えた。労せずして AI ラボは必要としていたソフトウェア・インフラのためのプログラムを手に入れた。ストールマンはオリジナルのヴァージョンになかった機能も付加して、プログ ラムをより強力にした。「私たちはそれを何年間も使っていました」とストールマンは言う。

1970 年代のプログラマの目から見れば、このやりとりは、工具や砂糖を借りるためにお隣に立ち寄るのと同じことを、ソフトウェアに置き換えたものだ。唯一の違い は、ストールマンが AI ラボのためにソフトウェアのコピーを借りて行ったために、誰かが元のプログラムを使えなくなる、なんてことがないことだ。何かあるとすれば、このプロセス で他のハッカーが得をする、ということだ。というのもストールマンはそのプログラムに彼自身の追加機能を導入したが、他のハッカーがそれをまた借りていく ことは歓迎されたからだ。たとえば、ストールマンは、そのプログラムを借りて行った、民間エンジニアリング企業 ボルト・ベラネック・ニューマン（Bolt, Beranek & Newman）社のプログラマのことを思い出す。彼は Twenex でそれが動くようにし、機能をいくつか追加した。ストールマンは最終的その機能を、AI ラボ自身のソースコード・アーカイヴに再統合した。二人のプログラマは共通ヴァージョンを一緒にメンテナンスすることにした。それはユーザーの選択により、ITS 上でも Twenex 上でも動作するコードを持つものだった。

「プログラムの成長は都市の成長に似ています。」AI ラボのソフトウェア・インフラを回想してストールマンは言う。「一部は置き換えられ、作り直されます。新しいものが追加されます。しかし、人はいつでもある部分を見て『ふーん、スタイルからすると、ここは 60 年代初期に書かれたんだね、こっちは 70 年代中頃のものだ。』と言えます。」

この知的累積の単純な仕組みによって、AI ラボや他の場所のハッカーたちは堅固な創造物を築き上げた。この文化に参加したプログラマすべてが自分をハッカーと呼んだわけではないが、多くがリチャー ド・M・ストールマンと同じ気分を共有していた。プログラムやソフトウェアのある改良が自分の問題の解決にそこそこ役立つなら、他の誰かの問題の解決に だって役立つものだ。カルマの好転を願って、それを共有しようじゃないか。

この協力システムは、商業的秘密主義と貪欲によって堀崩され、秘密と協力との奇妙な組み合わせになっていった。例えば、カリフォルニア大学バークレ イ校の計算機科学者たちは、AT&T から取得した Unix システムをもとにして、BSD と呼ばれる強力なオペレーティング・システムを作り上げた。バークレイはコピー用のテープ代だけで BSD を利用できるようにしていたが、それは AT&T から得た 50,000 ドルのソースコード・ライセンスを示すことができた学校に対してだけだった。バークレイのハッカーたちは、AT&T が許した範囲で最大限の共有を継続しようとしたが、彼らはこの二つの行為の間にある矛盾を認識していなかった。

同様に、ストールマンはゼロックスがソースコードファイルを提供しないことに悩まされていたが、まだ、怒ってはいなかった。彼はゼロックスにコピー を求めようとは全然思

わなかった。「すでにレーザープリンタをくれたのに、」とストールマンは言う。「我々に何かもっと借りがあるだろうとは言えませんで した。そのうえ、ぼくは、ソースコードをつけないのはある意識的な決定の反映であり、その変更を求めるのは無駄だと思っていました。」

やがて朗報が届いた。カーネギーメロン大学コンピュータ・サイエンス学部のある科学者がレーザープリンタのソースコードのコピーを持っているという内容だった。

カーネギーメロンと関わりがあるのは、良い兆しではなかった。1979 年、カーネギー・メロン大学の博士課程の学生、ブライアン・リード（Brian Reid）は、スクライブ（Scribe）と名付けた文書整形プログラムの共有を拒み、コミュニティにショックを与えていた。この文書整形プログラムは、 低レベルの書式指定の詳細（「この単語を斜体にする」とか「この段落の余白を狭める」とか）ではなく、やりたいことの意味（たとえば、「この単語を強調する」とか「この段落は引用文だ」とか）を指向したマークアップ・コマンドを持つ最初のものだった。共有するかわりに、リードはスクライブをピッツバーグ地 区のユニロジック（Unilogic）というソフトウェア会社に売却した。大学院生活も終わろうとしていたので、プログラムがパブリック・ドメインになら ないように骨折ってくれる開発者に委ねる方法を探していただけだと彼は言う。（パブリック・ドメインになるのが、なぜ望ましくないと特に考えたのかは不明 だ。） その取り引きの味つけに、リードはそのプログラムに時間依存機能、ソフトウェア・プログラマたちの言う「時限爆弾」をつけることにも同意した。それは無料 のコピー版を 90 日後に期限切れにし、動かなくするものだった。無効化を避けるために、ユーザーがソフトウェア会社に金を支払うと、会社は時限爆弾機能を解除する暗号を発行した。

ストールマンにとって、それは単純明快な、プログラマ倫理に対する裏切りだった。互いに共有するという考え方を尊重せず、リードは情報へのアクセス に対して、企業がプログラマに支払いを強要する手段を組み込んだ。しかし、スクライブをほとんど使わなかったので、ストールマンはその問題を深くは考えな かった。

ユニロジック社は、AI ラボに無料のコピーを提供したが、時限爆弾の除去も、それへの言及もしていなかった。それはしばらくの間動作した。そしてある日、一人のユーザーがスクラ イブが動かなくなったと言ってきた。システム・ハッカーのハワード・キャノン（Howard Cannon）は、そのバイナリを何時間かデバッグして時限爆弾を見つけ、パッチを当ててそれを除去した。キャノンは激怒し、わざと仕掛けたバグでユニロ ジック社に時間を浪費させられたことで頭にきている、と他のハッカーたちに語った。

ストールマンは、数ヶ月後、ラボの用事でカーネギーメロン大学のキャンパスを訪問した。訪問中、プリンタソフトウェアのソースコードを持っていると聞いた人物を探すのを忘れなかった。運良く、その男はオフィスにいた。

エンジニア同士の流儀で行われた会話は、友好的だが遠慮のないものだった。MIT からの来訪者であることを簡単に自己紹介した後、ストールマンは修正したいレーザープリン

タのソースコードのコピーを頼んだ。無念にも、研究者はそれを断った。
「コピーを渡さない約束なんです、と彼は言いました」とストールマンは言う。
記憶とはおかしなものだ。この事実から 20 年後、ストールマンの精神史のテープは所々空白になっている。ストールマンは、訪問目的や訪問がその年の いつなのかを思い出せないばかりか、会話の相手が誰だったかさえ思い出せない。 リードによれば、ストールマンの依頼に応対した人物は、ロバート・スプロウル（Robert Sproull）のようだ。ゼロックス・パロアルト研究所の元研究員で、現在は、計算機技術のコングロマリット、サン・マイクロシステムズ（Sun Microsystems）の研究部門、サン研究所の所長だ。1970 年代、ゼロックスのパロアルト研究所にいた頃、スプロウルは問題のレーザープリンタ 用ソフトウェアの中心的開発者だった。1980 年頃、スプロウルはカーネギーメロン大学で学部研究者のポストにつき、他のプロジェクトと並行してレーザー プリンタの仕事を続けていた。
しかし、この依頼のことを直接聞いてみると、スプロウルからは何も出て来ない。「事実関係についてはコメントできません」とスプロウルは電子メールに書いてきた。「その出来事は全く思い出せません。」
「ストールマンが求めたのは、スプロウルがカーネギーメロンに来る直前かその頃書いていた最新・最先端のコードだった」とリードは回想する。もしそ うなら、そこに誤解があったのかもしれない。ストールマンが欲しかったのは、MIT がすでに長い間使っていたプログラムのソースコードで、新しいヴァー ジョンではなかった。しかし、短い会話の中では、どのヴァージョンかは全く話題に上らなかった。
聴衆を前に話すとき、ストールマンは繰り返しこの出来事に言及し、この男がソースコードを渡したがらなかったのは、非開示契約のためだと指摘してき た。非開示契約は、ゼロックスとの間の合意契約で、守秘の約束と引き換えに、署名者にソースコードへのアクセスを認めるというものだ。今ではソフトウェア 産業の標準的なビジネス・アイテムだが、非開示契約すなわち NDA（nondisclosure agreement）は当時新しく開発されたもので、ゼロックスにとって、レーザープリンタの商業的価値とそれを動かすのに必要な情報の価値の両方を反映 していた。「当時、ゼロックスはレーザープリンタを商品化しようとしていました。」リードは回想する。「ソースコードを渡したとしたら正気じゃないでしょ う。」
しかし、ストールマンにとって、NDA は全く別の何かだった。それまでプログラムを共有資源とみなすようにソフトウェア・プログラマを後押ししてきた社会への参加を、カーネギーメロンの研究者が 拒否してきたのだ。何百年も使われてきた灌漑用水路が突然干上がってしまった農民のように、ストールマンが水路をたどってその源泉（source）までさ かのぼって行くと、ゼロックスのロゴ入りの新ピカの水力発電ダムがあった。
ゼロックスが仲間のプログラマたちを強要して、この秘密を強要する新システムへ参加させていたことを、ストールマンが実感するには、しばらく時間が かかった。最初は、個人的な文脈で断られただけだと思った。「ぼくはとても怒っていて、それをどう表現すればいいか分かりませんでした。そこで黙って向き を変え、外へ出ました、」とストールマンは

回想する。「ドアを乱暴に閉めたかもしれませんが、よく分かりません。覚えているのはそこから出て行きたかった ということだけです。協力してくれるものと思って彼のオフィスに行ったので、断られたらときにどう応じようかなんて考えていませんでした。彼に断られたと き、ぼくは失望し、怒るのと同時に、呆然として何も言えませんでした。」

それから 20 年たっても怒りは収まらず、それがこれまで出会った唯一の出来事ではなかったが、ストールマンはその出来事を、彼を倫理問題に直面させた出来事として描き 出す。次の数ヶ月間にストールマンと AI ラボのハッカーコミュニティにふりかかった一連の出来事に比べれば、遠く離れたカーネギーメロン大学での 30 秒間の緊張関係などささいなことに思える。にもかかわらず、ストールマンを集権的な権威に対して本能的に懐疑的な孤独なハッカーから、ソフトウェア開発の 世界に自由・平等・友愛の伝統的観念を適用する十字軍的活動家に変えた事件を整理する段になると、ストールマンはカーネギーメロン大学での出来事を特別注 目すべきものとして選び出してくる。

「あれは非開示契約との最初の遭遇でした。そして非開示契約には犠牲者があることを直接教えてくれたのです、」とストールマンは断言する。「この場合の犠牲者は、ぼくと研究所でした。」

ストールマンは後に説明している。「もし彼が個人的理由で協力を拒んだのなら、大した問題ではなかったでしょう。バカ野郎とは思ったかもしれません が、それだけのことです。彼の拒絶が個人的なものでなく、ぼくだけではなく誰とも協力しないことを事前に約束していたという事実が、これを大きな問題にし たのです。」

それ以前の出来事にも憤りは覚えていたが、とストールマンは言う、カーネギー・メロンの一件に遭遇することで、これらの出来事が彼がこれまで神聖視 してきた文化を侵し始めていることに気づいたのだ。彼は言う。「ソフトウェアは共有されるべきだという考えは前から持っていましたが、それについてどう考 えればいいのかは、はっきりしていませんでした。ぼくの考えは不明確で、それを世の中に簡潔な表現で伝えられるほどまとまってもいませんでした。この経験 をしてから、僕は何が問題で、それがいかに大きいものなのかを認識し始めました。」

世界的なエリート研究機関のエリート・プログラマとして、ストールマンは自分の仕事の邪魔にさえならなければ、同僚のプログラマが妥協や取引をして も完全に無視するつもりだった。ゼロックスのレーザープリンタが来るまで、ストールマンは他のコンピュータユーザたちがしかめっ面をして我慢しているマシ ンやプログラムを見下して満足していた。

しかし今や、AI ラボのネットワークの中にレーザープリンタが入り込んできたことで、何かが変わってしまった。紙づまりを除けば、機械は素晴らしく機能していたが、個人の好 みやコミュニティの必要に応じてソフトウェアを修正することはできなくなった。ソフトウェア産業の視点で見れば、このプリンタソフトウェアはビジネス戦術 上の変化を表していた。ソフトウェアは、ソースコード公開への要望を企業が受けつけないほど価値ある資産になっており、公開すれば潜在的な競争相手に安い 複製品を造るチャンスを与えることにな

るだろう。ストールマンから見れば、レーザープリンタはトロイの木馬だった。十年間の失敗の後、ユーザーが変更する ことも再配布することもできないソフトウェア、後のハッカーなら「占有的」（proprietary;プロプライエタリ）という用語を使うだろうが、そう いうソフトウェアが、こっそりと AI ラボ内部に地歩を得た。それは寄贈品を装ってやって来た。

ゼロックスが一部のプログラマに対して、守秘と引き替えに追加ギフトへのアクセスを提供していたことも腹立たしかったが、もっと若い頃にそんな取引 を提示されていたら、自分もゼロックスの申し出を受けてしまったかもしれないとストールマンは断っている。しかし、カーネギーメロンで遭遇したことへの怒 りは、ストールマンを道徳的無気力から立ち直らせる効果があった。それは、そうした将来への提案を懐疑的に見るのに必要な怒りを与えただけでなく、その状 況をじっくり眺めてみるように迫った。ある日仲間のハッカーがストールマンのオフィスに立ち寄ったとき、そのハッカーのソースコードへの求めを拒絶することが、突然自分の仕事になったとしたらどうだろう。

「ある人がそういうやり方ですべての同僚を裏切るように僕を誘ったとき、別の誰かがぼくとぼくのラボ全体にしたことに、ぼくがどれだけ怒ったかを思 い出しました。」とストールマンは言う。「だから、『この素敵なソフトウェア・パッケージを下さることには感謝します。でも、お求めの条件は承諾できない ので、ぼくはそれなしでやって行こうと思います。』と言いました。」

それは 1980 年代の喧騒の時代を通じて、ストールマンが心に抱いていた教訓だった。この十年間に MIT の同僚の多くが AI ラボを去り、自分たちの非開示契約書に署名した。彼らは、これは最高のプロジェクトに参加するための必要悪だ、と自分に言い聞かせたかもしれない。しか し、ストールマンにとって、NDA はプロジェクトのモラル的正当性を疑問視させるものだった。コミュニティにもたらされないのなら、技術的にエキサイティングなプロジェクトの何が良いの だ？

ストールマンはすぐに学ぶことになるが、そういう申し出の拒絶は個人的犠牲以上のものを意味した。守秘に関して同様な嫌悪感を共有していながら、そ の嫌悪感を道徳的により融通のきく方法で表現する傾向にあった同僚のハッカーたちから自分を隔てることになった。ストールマンはこう断じた。他人のソース コードへの求めに応じないのは、第二次世界大戦終結以来ソフトウェア開発を育んできた科学的使命に対する裏切りであるだけでなく、黄金律、自分が他人にしてもらいたいように他人にしなさい、という道徳の基本線への冒涜でもある。

だからレーザープリンタとそれがもたらした巡り合わせに重要な意味があるのだ。それがなかったら、もっと普通の人生を歩んでいたかもしれないとス トールマンは言う。商業プログラマの物質的な快適さと、目に見えないソフトウェアコードを書いて過ごす人生からくる究極の物足りなさとのバランスを取りな がらの。そこには透明感も、他人が取り組まない問題への懸命な取り組みもなかっただろう。最も重要なことは、義憤がなかっただろう。まもなく見るように、 その気持ちこそ、いかなる政治的イデオロギーや倫理的信条に

劣らず確実に、ストールマンの経歴を推進してきたのだ。

「あの日から、これは絶対に参加できない何かなのだと確信しました。」とストールマンは言い、便利さのために個人の自由を売り渡す行為（ストールマ ン的な非開示契約の説明）だけでなく、そういった本来倫理的に疑わしい取引を奨励する文化全体のことをほのめかした。「ぼくは、自分が犠牲になったように は、他人を犠牲にしないことに決めたんです。」

後注

1.「ハッカー」という言葉については、付録A『ハック、ハッカーズ、ハッキング』を参照。

第二章　2001 年ハッカーの旅

ニューヨーク大学のコンピュータ・サイエンス学部は、ウォーレン・ウィーヴァー・ホールの中にある。ワシントンスクエア公園の 2 ブロック東にある要塞のような建物だ。強力な業務用空調設備の排出口が、周囲に熱い空気の堀をつくり、散歩している人も忙しい人も等しくがっかりさせている。その堀を突破した訪問者は、もう一つの手強い関門に出会う。建物に一つしかない入口通路を入るとすぐにある、セキュリティのチェック・イン・カウンターだ。

そのセキュリティ検問所を過ぎると、雰囲気はいくらかリラックスする。とはいえ、セキュリティ対策のないドアやつっかいが外れた非常口の危険性を説く掲示が、一階のあちこちに散らばっている。全体として、こうした掲示は一つのことを思い出させてくれる。2001 年 9 月 11 日以前の比較的のどかなニューヨークでさえ、十分に注意深く、疑い深かったのだ。

掲示とは対照的に、ホール内のアトリウムにはどんどん訪問者が集まってくる。ニューヨーク大の学生らしき者も少しはいるが、大方は音楽ホールの外で メインの出演者を待ちわびて右往左往しているもじゃもじゃ頭のコンサート客のようだ。この朝のひととき、大衆がウォーレン・ウィーヴァー・ホールを乗っ取ってしまい、近くの警備員たちは、テレビでリッキー・レイクを見ながら、「その講演」について尋ねてくる訪問者たちに、講堂に向かって肩をすくめてみせるほかなかった。

講堂に入ると、訪問者は建物のセキュリティ手続きを一時停止させてしまった人物を見つける。その人物はリチャード・ストールマンだ。GNU プロジェクトの創設者、自由ソフトウェア財団の初代代表者、1990 年のマッカーサー奨学金受給者、米国計算機学会のグレース・マレー・ホッパー賞の受賞者（こちらも 1990 年）、2001 年の武田財団による武田賞共同受賞者、元 AI ラボハッカーである。GNU プロジェクト自身のサイト http://www.gnu.org を含む多くのハッカー関連のウェブサイトが告げていたとおり、ストールマンはかつてのホームタウン、マンハッタンにいる。マイクロソフト社による最近の反 GPL キャンペーンに反論する待望の講演をするためだ。

ストールマンの講演の主題は、自由ソフトウェア運動の歴史と未来だ。場所にも意味がある。マイクロソフトの上級副社長クレイグ・マンディー （Craig Mundie）がすぐ近くのニューヨーク大学スターン校経営学大学院に現れ、GNU 一般公衆ライセンス（GNU General Public License）、つまり、GPL を非難する講演をしてから、一月もたっていない。GPL は、ストールマンが 16 年前に最初に考案した法的装置だ。ソフトウェア秘密主義の波にストールマンが気づいたのは、ゼロックスのレーザープリンタのトラブルがあった 1980 年だった。コンピュータ産業を呑み込もうとしているソフトウェア秘密主義の波の高まりに対抗するために作られた GPL は、自由ソフトウェア・コミュニティの主要なツールへと進化していた。GPL は、簡単に言うと、著作権法の重みを使って、ソフトウェア・プログラム

に共有的な所有方式、最近の法学者が現在「デジタル・コモンズ」と呼んでいるものを 成立させる。GPL はこれを撤回不能にする。著作者がいったんこの方法でコミュニティにコードを与えれば、そのコードは他の誰かによって占有的コードに変えられてしまうこと はない。二次的著作物は、それがオリジナルのソース・コードの相当部分を含んでいれば、同じ著作権ライセンスを持つ必要がある。この理由から、GPL に批判的な人たちはこれを「ウィルス」ライセンスと呼び、それによって GPL は、それが触れた全てのソフトウェア・プログラムに広がっていくという、歪んだ印象を与えようとした。1

ますますソフトウェアに依存するようになり、ますますソフトウェアの標準規格の恩恵を受けるようになった情報経済の中で、GPL は諺にいうところの「棍棒」になった。かつて、それをソフトウェア社会主義だとして笑い飛ばした会社でさえ、その利点に気づくようになった。フィンランド の大学生リーナス・トーバルズ（Linus Torvalds）が 1991 年に開発した Unix（ユニックス）風のカーネル、Linux（リナックス）は、世界で最も人気のあるプログラミング・ツール、GNU Emacs、GNU デバッガー、GNU C コンパイラーなどと同じく、GPL でライセンスされている。これらのツールをより集めて自由ソフトウェアのオペレーティング・システム GNU/Linux ができあがっており、世界中のハッカー・コミュニティによって開発され、養育され、所有されている。このコミュニティを脅威と見るかわりに、IBM、 ヒューレット・パッカード、サン・マイクロシステムズのようなハイテク企業はこのコミュニティをあてにするようになり、この常に成長し続ける自由ソフトウェアのインフラ上に構築されたソフトウェア・アプリケーションやサービスを販売している。2

彼らはマイクロソフト（Microsoft）に対するハッカーコミュニティの積年の戦いの戦略的な武器としての GPL にも頼るようになった。マイクロソフトは、ワシントン州レドモンドに本社を置き、1980 年代後半以降パソコンのソフトウェア市場を支配してきた会社だ。人気の Windows（ウィンドウズ）・オペレーティング・システムのオーナーとして、マイクロソフトは、産業規模で GPL ライセンスへの移行が起きれば最も損害を受ける立場にある。巨人 Windows のどのプログラムも、実行プログラムの占有的地位を宣言した著作権と契約（エンドユーザー・ライセンス契約つまり EULA）によってカバーされている。ソースコードも同様だが、いずれにしてもユーザーはそれを入手できない。これらのプログラムの中に「ウィルス的な」 GPL で守られたコードを組み入れることは禁じられている。GPL の要求事項を満たすためには、マイクロソフトはプログラム全体を自由ソフトウェアにすることを法的に求められることになる。そうすると、ライバル会社は、 コピー、修正、改良版の販売をすることが可能になり、マイクロソフトからユーザーを閉じ込めておくための基盤を取り去ることになる。

それゆえ、マイクロソフトは GPL の採用率をますます気にしているのだ。それゆえ、最近の講演でマンディーは、GPL やソフトウェア開発と販売に関する「オープンソース」のアプローチを非難したのだ。（マイクロソフトは「自由ソフトウェア」という用語を認めて

おらず、攻 撃の矛先を、第 11 章で描くように非政治的で、自由ソフトウェア運動とは距離を置いている「オープンソース」陣営に向けることを好む。）そしてそれゆえ、ストールマンは、マ ンディーの講演に対する公開の反論を、今日、同じキャンパスですることにしたのだ。

ソフトウェア産業では、20 年は長い年月だ。次のことを考えてみればよい。1980 年にリチャード・ストールマンが AI ラボのゼロックスのレーザープリンタを呪っていた頃、現在世界規模でソフトウェア産業を支配しているマイクロソフトは、まだ株式を公開していないスタート アップだった。IBM は、当時コンピュータ・ハードウェア産業で最強の勢力とみなされていた会社だが、現在の低価格パソコン市場に火をつけた最初のパーソ ナル・コンピュータはまだお披露目していなかった。今では私たちが当然のことと思っている多くの科学技術 -- ウェブ、衛星放送、32 ビットゲーム機 --は存在さえしていなかった。AOL、サン・マイクロシステムズ、アマゾン、コンパック、デルといった会社、数え上げればきりがないが、法人企業の上位 を占める更に多くの会社についても同じことが言える。

自由より進歩を評価する人々の間では、先端技術市場がこれほど短期間に拡大した事実が、GPL を肯定する側からも反対する側からも援用される。或る者は GPL 肯定論から、大半のコンピュータ・ハードウェアのプラットホームの寿命の短さを指摘する。時代遅れの製品を買わされるリスクに直面して、消費者は長期間生き残る最善の会社に群がる傾向がある。その結果、ソフトウェア市場は勝者総取りの闘技場になっている。3 占有的ソフトウェア環境は、独占の乱用と停滞につながる、と彼らは言う。強い会社が、ライバルや革新的な新設企業に代って市場の酸素をみんな吸い取ってし まうのだ。

他方は正反対のことを論じる。ソフトウェアの販売は、ソフトウェアの購買よりリスキーでないにしても、同程度にはリスキーだ、と彼らは言う。私的に 所有している「キラー・アプリ」（つまり、全く新しい市場を離陸させる現状打破的なテクノロジー）の経済的な見返りへの期待はいうに及ばず、4 占有的ソフトウェア・ライセンスが提供する法的保証なしでは、企業は参加する気が失せてしまう。またしても、市場は停滞し、技術革新は衰退して行く。同じ キャンパスの 5 月 3 日の講演でマンディー自身が述べているように、自社ソフトの独自性を競争的資産としているいかなる会社にとっても、GPL の「ウィルス的な」性質は「脅威となる」。マンディーは次のように言い足している。

「それは独立した商業的ソフトウェア・セクターを根本的に掘り崩すのです。なぜなら、受領者が製品に対して流通のコスト以上のものを支払うことを基礎にしているソフトウェアの流通を実質的に不可能にしてしまうからです。」5

この 10 年間、GNU/Linux と Windows が共に成功していることは、この問題に関するどちらの立場も、正しい場合があることを示唆している。しかし、ストールマンのような自由ソフトウェアの活動 家は、それを枝葉の問題と考えている。本当の問題は、と彼らは言う、自由ソフトウェアと占有的ソフトウェアのどちらが、より成功するかではない。どちらが より倫理的かだ。

だが、勢いを求めて戦うことは、ソフトウェア産業では重要だ。たとえばマイクロソフトのように強力な販売元も、第三者のソフトウェア開発者の支援に 依存にしている。第三者のソフトウェア開発者のツール、プログラム、ゲームが、Windows のような土台になるソフトウェアを、主流の消費者にとってより魅力的にするのだ。ここ 20 年間の科学技術市場の急速な進化を例に挙げつつ、もちろんこの間の自社の目を見張る業績は言うまでもないが、マンディーは聴衆に、自由ソフトウェア運動の 最近の勢いに夢中になり過ぎないように、こう助言した。

「20 年間の経験は、知的財産を守り、研究開発費用を取り戻せるビジネスモデルが、印象深い経済的利益とその利益の非常に広範囲にわたる分配を可能にすることを示してきたのです。」6

そんな忠告も、ストールマンの今日の講演の背景として役立っている。彼らの発言から一月もたたないうちに、ストールマンは部屋の正面の黒板の一つに背を向けて立ち、開演を前にして神経質になっている。

最近の 20 年間がソフトウェア市場に劇的な変化をもたらしたとすれば、その年月はストールマン自身にさらに劇的な変化をもたらした。愛する PDP-10 と一日中語り合っていた、やせて、きれいにひげを剃ったハッカーはもういない。そのかわりに今では、長髪でラビ風のひげをはやした中年のずんぐりした男 が、電子メールをやりとりし、仲間のプログラマに熱弁をふるい、今日のような講演をすることに自分の時間の大半を使っている。水色のＴシャツを着、茶のポ リエステルのズボンをはいたストールマンは、まるで救世軍の着替室から出てきた砂漠の隠者のようだ。

聴衆は、ストールマンのファッションと身だしなみの趣味を共有する訪問者でいっぱいだ。ラップトップ・コンピュータと無線通信用モデムを持っている 人が多いが、それはインターネットで待ち受けている聴衆に、ストールマンの言葉を記録して送るのにうってつけだ。男女比はおよそ男性 15 人に対して女性 1 人、部屋の中の女性 7、8 人に 1 人は Linux の公式マスコットのペンギンのぬいぐるみを抱えている。ほかのぬいぐるみはテディーベアだ。

落ち着かないストールマンは部屋の正面の持ち場を離れ、最前列の椅子に座り、開いてあったラップトップに向かってコマンドを叩く。それから十分間、ストールマンは講堂のステージの足下にいる彼を取り囲む、学生や教授やファンの数がどんどん増えていくことに気がつかない。

講演が始まる前に、アカデミックな形式の仰々しい儀式を見なければならない。ストールマンの登場は紹介者が一人では足りず、二人分に値する。最初の紹介者は、スターン校の先端技術センターの共同所長、マイク・ウレツキー（Mike Uretsky）だ。

「大学の役割は、論争を促し、興味深い議論をすることです。」とウレツキーは言う。「とりわけこの講演会、このセミナーはまさにそういった性質のものです。オープンソースの議論は特に興味深いと思います。」

ウレツキーが言葉を継ぐ前に、ストールマンが動かなくなった自動車の運転手のように手を振って立ち上がる。

「私がしているのは自由ソフトウェアです、」ストールマンは湧き上がる笑い声に向かって言う。「オープンソースは別の運動です。」

笑い声はやがて拍手喝采になる。この部屋にはストールマンの味方がズラッといる。彼らは言葉に厳密なストールマンの評判を知っている人々だ。大いに 評判になった 1998 年のオープンソース・ソフトウェアの提唱者との仲違いを知っていることは言うまでもない。大部分は、こういう爆発を期待して来たのだ。昔、ラジオファン が、番組でジャック・ベニーのトレードマークの台詞「もう、やめて("Now cut that out!")」を毎回待っていたのと同じことだ。

ウレツキーは慌てて紹介を終え、演台をエドモンド・ショーンバーグ(Edmond Schonberg)に譲る。ショーンバーグはニューヨーク大学のコンピュータ・サイエンス学部の教授である。コンピュータ・プログラマとして、また GNU プロジェクトへの貢献者として、ショーンバーグは踏んではいけない言葉の地雷をわきまえている。彼は現代のプログラマの視点でストールマンの経歴を巧みに 要約する。

「リチャードは、ローカルに行動することでグローバルに思考を始めた人物の好例です。問題はソースコードの入手不能に関係していました。」ショーン バーグは言う。「彼は、首尾一貫した哲学を発展させ、私たち全てに、ソフトウェアはいかにして生み出されるのか、知的財産とは何を意味するのか、そして、 ソフトウェア・コミュニティが実際に何を表現しているのか、といったことについて私たちが持つ観念を再検討させてきました。」7

ショーンバーグがストールマンに歓迎の意を表すると、拍手はさらに大きくなる。ストールマンはラップトップを止めるのに少し時間をとり、椅子から立ち上がり、舞台に上った。

最初、ストールマンの講演は、政治的演説というよりキャッツキルス喜劇のネタのようだ。「わたくしは、この壇上に上がる機会を提供して下さったマイ クロソフトに御礼を申し上げたい。」とストールマンは辛辣だ。「この何週間か私は、偶然どこかで発禁処分をくらった本の作者みたいな気分でした。」

初心者のために、ストールマンは、まず自由ソフトウェアについての助走的なたとえ話から始める。彼はソフトウェアのプログラムを料理のレシピにたとえる。プログラムも料理のレシピも、提供するのは、望んだ仕事を完成するのに役立つ方法を一歩毎に指示したものだ。そして特にユーザーが希望していること や特別な事情があれば、簡単に変更することもできる。「レシピ通りにしなくてもいいんです、」とストールマンは注意する。「料理の具は減らしても構いませ ん。きのこ好きなら、きのこを足してもいいし、医者にとにかく塩分を減らすように言われているなら、塩は減らしましょう。」

一番大切なことは、ソフトウェアのプログラムとレシピは、どちらも簡単に共有できることだ、とストールマンは言う。ディナーの客にレシピをあげると きに、料理人が失うのは紙代とレシピを書く時間くらいのものだ。ソフトウェアのプログラムはさらに失うものが

少ない。通常は数回のマウスをクリックと少し の電気だけだ。しかし、どちらの場合にも、情報を与える人は二つのものを得る。友情の深まりと、お返しに面白いレシピを借りるられるようになること、と。

「想像してください。レシピがブラックボックスの中に包装されていたら、どうなるでしょう。」とストールマンは言い、ギアを入れ変える。「何を材料 に使っているのか分からず、ましてレシピを変えることはできません。で、友達にレシピのコピーをつくってあげたら、どうなるでしょう。彼らは、あなたを海 賊と呼び、刑務所に何年間もほうりこんでやろうとするでしょう。そんな世界は、レシピを共有していたすべての人々を憤慨させるでしょう。しかし、それがまさに占有的ソフトウェアの世界の有様なのです。その世界では、他人に対する普通の良識が、禁じられたり、妨害されたりするのです。」

この初心者へのたとえ話が終わると、ストールマンはゼロックス・レーザープリンタのエピソードの再話にとりかかる。レシピのたとえ話と同様、レーザープリンタの物語は手頃な説得手段だ。その寓話風な組み立てで、ソフトウェアの世界では物事がいかに速く移り変わってしまうものなのかを劇的に見せてく れる。聴衆を、アマゾンのワンクリックショッピング、マイクロソフト Windows、オラクル・データベース以前の時代に連れ戻し、今のように法人企業のロゴのついていない、ソフトウェア所有権の観念を検討してみることを 聴衆に求める。

ストールマンは最終弁論をする地方検事のやり方をそっくりまねて物語る。カーネギーメロン大学の教授がプリンタのソースコードを貸すのを拒絶したところに差し掛かると、ストールマンは一瞬沈黙する。

「私たちを裏切ったのです。」とストールマンは言う。「しかし彼は、私たちを裏切っただけではありません。場合によっては、あなたのことも裏切ったでしょう。」

「あなた」と言いながら、とがめるようにストールマンは予期していなかったある聴衆に人差し指を向ける。狙われた聴衆はひるんで眉がピクリと動く が、ストールマンの目はもう次に移っている。ゆっくりと慎重に二人目の聴衆を選び出すと、聴衆から忍び笑いが起こる。「思うに、おそらく彼はあなたに対してもそうしたでしょう。」と彼は言い、最初に選んだ聴衆の三列後ろにいる聴衆を指さす。

ストールマンが三人目の聴衆を選ぶ頃には、忍び笑いは大笑いに変わっている。彼の仕草は少し芝居がかっているが、それは実際にそうだからだ。ゼロッ クスのレーザープリンタの話をしめくくるときになっても、やはり、ストールマンは芸人並に大げさだ。「たぶん彼は、この部屋にいる大部分の人に対してそう したでしょう。1980 年にまだ生まれていなかった人は別ですが。」とストールマンは言い、さらに笑いを取る。「彼は地球上のほぼ全員と助け合うのを拒否するような約束をしていたのです。」

ストールマンはコメントの意味が理解されるように半拍あける。「彼は非開示契約にサインしていたのです。」とストールマンは言葉を継ぐ。

リチャード・マシュー・ストールマンが挫折した研究者から立ち上がり、この 20 年間で

政治的リーダーとなったことは多くのことを物語っている。それはストールマンの粘り強さと並外れた意志の力を物語っている。それはストールマンが作るのを助けた自由ソフトウェア運動の、明確に組み立てられた構想と価値を物語っている。それはストールマンが作った高品質なソフトウェア・プログラムを物語っている。それらのプログラムはプログラミングの伝説的人物としてのストールマンの名声を確立した。それは GPL の増大していく勢いを物語っている。ストールマンを見てきた人の多くが、GPL という法的発明品を、彼の最も重要な功績と見ている。

最も重要なことはそれが、コンピュータ技術とその技術に力を与えるソフトウェア・プログラムにますます恩恵を受けるようになった世界の中で、政治的力の性格が変化してきていることを物語っていることだ。

おそらくそれが理由なのだろう。大部分のハイテクの星たちが衰退しつつあるときでさえ、ストールマンの星は輝きを増してきた。1984 年に GNU プロジェクトを開始して以来、8 ストールマンは節目ごとに無視され、揶揄され、けなされ、自由ソフトウェア運動の内外から攻撃されてきた。その全てを通じて、GNU プロジェクトは、有名な立ち遅れはいくつかあったものの、何とか里程標にたどりつき、18 年前に入り込んだときより桁違いに複雑になっているソフトウェア市場で存在感を保っている。自由ソフトウェアというイデオロギーも同様だ。ストールマン自 身が手塩にかけてきたイデオロギーだ。

この通貨の背景理由を理解するには、彼自身の言葉によるだけでなく、これまで彼と協力したり争ったりしてきた人々の言葉によっても、リチャード・ス トールマンを検討してみることが役に立つ。リチャード・ストールマンの性格の素描は複雑ではない。「あなたの見たものが、あなたの得るものだ」 ("what you see is what you get") という古来の格言を体現した人物が、リチャード・ストールマンだ。

「リチャード・ストールマンという人間を理解したいなら、部分の全てを一貫した全体として見る必要がある、と私は思っています。」と自由ソフトウェ ア財団の法律顧問でコロンビア大学ロースクールの法学教授、エベン・モグレン（Eben Moglen）は助言する。「多くの人がストールマンを知るときの障害だと見ている変人ぶり全てが、まさにストールマンなのです。リチャードの強い挫折 感、原理的なまでの道徳的責任感の強さ、非妥協性、とくに彼が根本的だと思っている問題に対する非妥協性。まさしくこれらすべてが、リチャードがいつ何を したのかの理由なのです。」

レーザープリンタから始まった旅が、やがて世界屈指の金持ち企業とのボクシングマッチへと導いていった経緯を説明するのは容易な仕事ではない。それ には、今日の社会でソフトウェア所有権を非常に重要なものにした諸力を思慮深く検討することが求められる。また、彼以前の多くの政治指導者たちのように、 人間の記憶の影響されやすさを理解している一人の人物を思慮深く検討することが求められる。ストールマンの周囲に長い時間をかけて築き上げられた神話や政 治的な色合いの婉曲表現を翻訳する能力が必要である。最後に、ストールマンのプログラマとしての天分と、その才能を他のことの追求に変えること

への彼の成 功と失敗を理解する必要がある。
彼自身がこの旅を要約するとき、ストールマンはモグレンが観察した個性と信条の融合を認める。「頑固さはぼくの長所だ」と彼は言う。「何か大きな困難を伴うことをしようとするとき、たいていの人は結局、心がくじけて断念してしまいます。ぼくは決して諦めません。」
彼は偶然のおかげだともいう。ゼロックスのレーザープリンタのもめ事がなく、MIT の職員のキャリアを終えることになった個人的・政治的な軋轢がなく、その他半ダースほどのタイムリーな要因がなければ、別の職歴を歩んでいただろう自分の人生をストールマンは容易に思い浮かべることができる。とはい え、ストールマンはその違いを生んだ立場に彼を置いた諸力と環境に感謝している。
「ぼくはちょうどふさわしい技能を持っていたんです、」とストールマンは、GNU プロジェクトを立ち上げた時の決意を聴衆に要約してみせる。「他には誰もいませんでした。そこで、こんな風に感じたんです。『ぼくは選ばれたんだ。この仕 事をしなくちゃ。ぼくがしなかったら他の誰がするんだ。』」

後注

1 実際には、GPL にはそんな力はない。同じコンピュータの中で、あなたのコードを GPL がカバーするプログラムの中に置くだけでは、コードを GPL の下に置くことにならない。「何かをウィルスと比較するのはとても酷なやり方です、」とストールマンは言う。「オリヅルランと較べる方が正確です。能動的 に挿し木すれば他の場所にも移植できるんです。」 GPL に関する追加情報は、http://www.gnu.org/copyleft/gpl.html にある。

2 これらのアプリケーションが、GNU/Linux 上で走っているからといって、それら自体が自由ソフトウェアだということにはならない。むしろ、それらの大半は占有的ソフトウェアで、Windows 以上にあなたの自由を尊重してくれているわけではない。それらは、GNU/Linux の成功に寄与するかもしれないが、その存在理由である自由という目標に寄与するわけではない。

3 Shubha Ghosh, "Revealing the Microsoft Windows Source Code," Gigalaw.com (January, 2000), http://www.gigalaw.com/ を参照。

4 キラーアプリは占有者的である必要はない。しかし読者は要点をつかんでいると思う。ソフトウェア市場は宝くじに似ている。払戻金が大きければ大きいほど人 はそれが欲しくなる。キラーアプリ現象の良い要約は、Philip Ben-David, "Whatever Happened to the `Killer App'?" e-Commerce News (December 7, 2000) http://www.ecommercetimes.com/perl/story/5893.html を参照。

5 マイクロソフト社上級副社長であるクレイグ・マンディーの「商用ソフトウェア・モデル」("The Commercial Software Model")を参照。2001 年 5 月 3 日のニューヨーク大学スターン校経営学大学院でのマンディーの講演のオンライン記録から抜粋。
http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp

6 同上。

7 今日これを言ったら、ストールマンは偏りと混乱をもたらすものとして「知的財産」という用語に異議を唱えただろう。次を参照。 http://www.gnu.org/philosophy/not-ipr.html

8 頭字語 GNU は、"GNU's not Unix."(GNU は Unix にあらず。)を表している。2001 年 5 月 29 日のニューヨーク大学の講演の別の個所でこの頭字語の由来を次のように要約している。

私たちハッカーはいつでも、プログラムに愉快で、いたずらっぽい名前をつけようとします。というのも、プログラムを書く楽しみの半分はプログラムに名前を付けるときにあるんですね。私たちには再帰的頭字語を使うという伝統もあります。そ れによって、今書いているプログラムが、何か既存のプログラムと似たものだと言いたいわけです. . . 。そこで、(Something Is Not UNIX ほにゃららは Unix にあらず。)に対する再帰的頭辞語を探しました。26 文字全て試しましたが、該当する語はありませんでした。私は縮めてみることにしました。 Something's Not UNIX に対する 3 文字の頭字語ならあるかもしれません。1 文字ずつ試していくと、GNU という単語に行きあたりました。」そういうことです。

ストールマンは駄洒落の愛好家だが、ソフトウェア・ユーザーに頭字語の最初の"g"を発音することを推奨する(つまり、"gah-new" ガー・ヌー)。こうするとアフリカのカモシカの仲間 Connochaetes gnou(オジロヌー)の名前 "gnu"(ヌー)との混同を避けられるし、形容詞 "new" ([ヌー/ニュー; "news" をヌーズとなまっていた? "new" は "njúː" で、グヌーにもグニューにもなっていたかに読める。]) との混同も避けられる。「今まで 17 年間これでやってきているからもう、全然新しく("new") ないですがね。」とストールマンは言う。

出典は、2001 年 5 月 29 日のニューヨーク大学でのストールマンの講演、"Free Software: Freedom and Cooperation"(自由ソフトウェア:自由と協力) のオンライン記録と著者の注記。 http://www.gnu.org/events/rms-nyu-2001-transcript.txt

第三章　若いハッカーの肖像

リチャード・ストールマンの母、アリス・リップマンは息子の特別の才能に気付いた瞬間を今でも憶えている。

「8 歳のときだったと思います。」とリップマンは回想する。

年は 1961 年、離婚して間もないシングル・マザーのリップマンは、マンハッタンのアッパー・ウェスト・サイドにあった一家のワンベッドルームしかない小さなアパートで、週末の午後を何をするということもなく過ごしていた。サイエンティフィック・アメリカン誌をめくっていくと、リップマンのお気に入りの箇所が出てき た。マーティン・ガードナーが執筆していた「数学ゲーム」というコラムだ。当時、美術の臨時教員をしていたリップマンは頭の体操にガードナーのコラムを楽 しんでいた。リップマンは以前から近くのソファで本に没頭していた息子と、その週の目玉のパズルをやってみることにした。

「パズルを解くのは得意じゃないんです。」と彼女は認める。「でもアーチストとして、それは概念的な障害物を通リ抜けていくのに役に立つことがわかりました。」

パズルを解こうとする試みはすぐ壁にぶつかってしまった、とリップマンは言う。嫌になって雑誌を投げ出そうとしたとき、シャツの袖を優しく引っ張られてびっくりした。

「リチャードでした。」彼女は回想する。「助けてほしいか知りたがっていました。」

パズルと息子を何度も眺めて、その申し出に最初は懐疑的だったとリップマンは言う。「リチャードに雑誌を読んだのって聞きました、」と彼女は言う。「読んだよって彼は言うんです。読んだどころか、もうパズルを解いていました。次に、彼はどうやって解くのかを説明し始めました。」

解法のロジックを聞きながら、リップマンの最初の懐疑はすぐに信じ難い思いに変わった。「頭のいい子だとはいつも思っていました、」と彼女は言う。「でも、彼がどれだけ進んでいるかを示すものを見たのは、これが初めてでした。」

このでき事から 30 年後、リップマンはこの思い出話を笑いで中断する。「正直に言えば、私にパズルの解き方が理解できたとは思いません、」と彼女は言う。「覚えているのは、リチャードに答えが分かったのに驚いたことだけです。」

二度目のマンハッタンのアパートは、同じ大きさの広々とした 3 ベッドルームの組み合わせで、今は亡きモーリス・リップマンと 1967 年に結婚した後、彼女と息子はそこに引っ越してきた。そのアパートのダイニングルームのテーブルの前に座って、小さい頃の息子を思い出すとき、アリス・ リップマンから滲み出てくるものには、ユダヤ人の母親の誇らしさと困惑とが混ざり合っている。近くの食器棚には、髭面で博士の礼服を着こみ、睨みつけるよ うな顔をしたストールマンの 8x10 インチの写真が飾ってある。その肖像のせいで一緒に置かれたリップマンの甥と姪の写真が小さく見える。しかし、訪問者 がそれを言う前に、その目立つ配置に対して皮肉を言うことで、リップマンはバランスをとる。

「グラスゴー大学から名誉博士号を授与されてから、その写真を私が持っているようにと

リチャードが言うんです、」とリップマンは言う。「『何でだかわかる、ママ？　ぼくが出席した初めての卒業式だよ。』ですって。」

そういうコメントは神童を育ててきたことからくるユーモアのセンスの表れだ。息子の頑固さや特異な行動について読んだり聞いたりしたことに対して、リップマンはお返しに、少なくともその十倍の逸話を提供できることは確かだ。

「リチャードはとっても保守的だったの、」憤慨したかのように両手をあげ、リップマンは言う。「まさにこのテーブルでいつも最低の議論をしていまし た。私は市立学校の教員組合を結成しようとしてストライキをした最初のグループの一人でした。それで、リチャードは私にとっても怒っていました。組合は堕 落だと思っていたんです。それから社会保障にも大反対でした。人はお金を自分で投資することで、もっとずっと多くの金を稼げるんだと考えていました。10 年もしないうちに、あれだけ理想主義的になるなんて誰が思ったでしょう。そういえば、彼の義理の姉が私のところにきて『大きくなったら何になるんでしょうね。ファシストかしら。』と言ったものです。」2

リチャードの父、ダニエル・ストールマンとリップマンは、1948 年に結婚して 1958 年に離婚し、その後は共同親権だった。リップマンは約 10 年間シングルで親をしていた者として、息子の権威嫌いについて証言できる。彼女はまた息子の知識欲についても証言できる。そのふたつの力が組み合わさった とき、彼女と息子は最大のバトルをしていたのだ、とリップマンは言う。

「まったく食事をとりたがらないことがありました。」と、8 歳ぐらいに始まって 1970 年に高校を卒業するまで終らなかった行動パターンを思い出しながらリップマンは言う。「夕食に呼んでいるのに、聞こえないんです。気づかせるために、9 回も 10 回も、呼ばないといけませんでした。完全に没頭していたんです。」

ストールマンも同じ状況を思い出すが、その記憶には少し政治的なひねりが加わっている。

「本を読むのが楽しかったんです。」と彼は言う。「ぼくは本を読みたい、母はぼくにキッチンへ行って食事をして寝るようにと言う。そんなときには、 耳を貸さないことにしました。本を読んじゃいけない理由は分からなかったし。当時、何でぼくにどうしろと言えるのかが分からなかったし。基本的には、民主 主義とか個人の自由の概念について読んでいて、自分自身に当てはめてみました。子供がそういった原理から締め出される理由は見当たらなかったですね。」

専制的な権威に対する個人の自由の信念は、学校でも同じだった。11 歳のときにはクラスメートよりも 2 年進んでいたストールマンは、才能のある公立学校生徒によくある欲求不満のすべてに耐えていた。彼の母親が、教師と親の長い面談に初めて行ったのは、あの パズルの出来事から間もない頃だった。

「作文を絶対に書こうとしませんでした。」最初の頃の物議を回想してリップマンは言う。「高校の最上級生になる前に書いた最後の作文は、4 年生の教師のために書いた西洋の記

数法の歴史についてのエッセイだったと思います。」実際に書きたいことが何もないときに特定の話題を選んで作文を求めら れても、ストールマンにとって、それはほぼ不可能なことだったし、そんな状況を避けるためなら何でもするぐらい、十分に苦痛だった。

分析的思考が要求されることには何にでも才能がめぐまれていたストールマンは、他の学科を犠牲にして数学と科学に引き寄せられていった。しかし、一 部の教師には一心不乱な様子に見えているものが、リップマンには我慢が足りないものに見えていた。とくに彼女の息子が自然に惹き付けられることの少ないよ うにみえる主題や趣味と比較すると、数学と科学は学ぶ機会が多く提供されているだけのことだった。10 歳か 11 歳のころ、ストールマンのクラスの男の子たちは、タッチフットボールの公式戦に参加し始めるようになった。彼女は息子が怒って帰ってきたのを覚えている。 「プレーはとってもしたかったんだけど、釣り合うだけの技能は身につけてません。」「それで彼はすごく怒ってました。」

結局、彼女の息子はその怒りでますます数学と科学に集中することになった。とはいえ、科学の領域でさえ、彼の短気は問題だった。ストールマンは 7 歳までに微積分の教科書を熟読して、大人には話を噛み砕いて聞かせてやる必要がほとんどないと思った。中学のいつだったか、リップマンが息子の兄貴分とし て遊んでくれるようにコロンビア大学の学生を雇った。その学生は、1 回目で家族のアパートメントを後にして、もう 2 度と戻ってこなかった。「リチャードが話したことがちんぷんかんぷんだったんだと思うわ」と、リップマンは推測している。

1960 年代初めまでさかのぼって、母として別のお気に入りの思い出がパズルの件の少し後にある。7 歳の頃、離婚してクイーンズから転居して 2 年後、リチャードはリバーサイド・ドライブ・パークのすぐ隣でロケットを打ち上げる遊びに興味を持ち始めた。目的のない遊びだったものが、発射ごとにデー タを記録し始めてから、真剣な興味の最先端に変わった。数学ゲームへの興味と同じで、その研究はある日まで、つまり NASA の大きな打ち上げの直前の日まで、あまり注意をひかなかった。リップマンは、打ち上げを見たいのかどうか聞いてみた。

「彼は煙を出してました。」リップマンは言う。「彼が私に言えたのは、『でも、ぼくはまだ発表してない。』明らかに、NASA に何か知らせたいと本気で思っていました。」ストールマンは、この出来事を覚えていない。しかし、ありそうな事は、何も示せるものがないことに苦しんでい たんじゃないのかと言う。

そんな逸話は、ストールマンの人生を貫くトレードマークになる集中性が、早いうちに表れていたことを物語る。他の子供たちがテーブルに着いたとき も、ストールマンは自分の部屋で本を読んでいた。他の子供たちがジョニー・ユナイタスごっこをしていたときも、ストールマンはウェルナー・フォン・ブラウ ンごっこをしていた。「ぼくは変わっていたんだな。」ストールマンは、1999 年のインタビューで、小さいころの自分を簡潔に要約した。「ある年齢を境に、ぼくの友達は教師だけになってしまいました。」3 ストールマンは自分の風変わりな個性を恥じていなかったし、彼が失敗とみなしていた社会的不適応と区別し

ていた。しかし、どちらも彼の社会的疎外に貢献し た。

それは学校での揉め事がふえることも意味していたが、リップマンは息子の情熱をかなえてあげることに決めた。12 歳までに、リチャードは、夏はサイエンス・キャンプに参加し、学校は私立学校に行くことになった。ある教師がコロンビア・サイエンス・オナーズ・プログラ ム（the Columbia Science Honors Program）に入れることを勧めた。これはニューヨーク市の才能に恵まれた中高生のためのスプートニク以来のプログラムだ。ストールマンの課外活動が ふえて、まもなく毎週土曜日はコロンビア大学のキャンパスへ通うようになった。

コロンビア・サイエンス・オナーズ・プログラムでフェロー・クラスメートだったダン・チェス（Dan Chess）は、リチャード・ストールマンを、数学と科学への同じ欲求を分かち合っていた生徒の中でさえ、ちょっと変なやつに見えたと回想する。「ぼくた ちはみんな、おたく（geeks and nerds）だったけど、そのなかでも異常に適応がへただった。」現在、ハンター・カレッジの数学教授をしているチェスは回想する。「彼は非常に頭がよ かった。頭のいい人はたくさん知っているけど、ぼくの知る限り、彼は一番頭がよかったね。」

コロンビア・サイエンス・オナーズ・プログラムの同窓のフェローで、SF と SF 大会への情熱を分かち合っていたためストールマンと付き合いのあるコンピュータ・プログラマの、セス・ブライドバート（Seth Breidbart）は、それを裏付ける証言をしている。彼は 15 歳のスポーツ刈りのストールマンを「怖ろしい奴」と回想する。特に 15 歳のフェローにしては、だ。

「表現するのが難しいね」とブライドバートは言う。「近寄り難かったわけじゃない。けど、まさに強烈だった。とても博識だったけど、いろいろな意味で頭の固いところもあった。」

そういう表現は、次のような推測を生む。「強烈」とか「頭の固い」といった判断を含む形容詞は、今日、少年期の行動障害として分類されている特性の 形容にすぎないのではないか。2001 年 12 月のワイアード・マガジンの「ギーク・シンドローム」というタイトルの記事は、高機能自閉症とかアスペルガー症候群と診断された科学的才能に恵まれた何人かの子供の人物像を描いている。ワイアードの記事に記録された親たちの回想と、リップマンが話してくれたそれとは、いろいろな意味で不気味なほど似てい る。ストールマンさえ、精神医学的な見直しに時々ふけっていた。トロント・スターに記載された 2000 年時点のプロフィールでは、ストールマン自ら、「自閉症の境界線上」とインタビュアーに描写していた。4

そういう推測は、もちろん、昨今のいわゆる「行動障害」と呼ばれるもののほとんど当てにならない性質から生じている。「ギーク・シンドローム」の著 者スティーブ・シルバーマンは、アメリカの精神科医たちはアスペルガー症候群をつい最近になって幅広い行動特性をカバーする包括的用語として有効だと認め たにすぎないと書いている。その特性の範囲には、貧弱な運転技術や貧弱な社会的適応から、高い知性、数字やコンピュータや秩序だったシステムの偏愛までが 含まれている。5

「それに似た何かがあったということも、ありえます。」と彼は言う。「でも一方、その症候群の特徴の一つはリズムについていくのが困難なことですけ れども、ぼくはダンスができる。実際、すごく複雑なリズムについていくのが好きです。十分にわかったと言えるほどはっきりした話じゃないんですね。」他の 可能性は、ストールマンが「シャドウ・シンドローム」だったというもので、それはアスペルガー症候群と共通の方向にあるが、正常の範囲を越えないものだ。 6

チェスは、彼もその一人ということになるが、そんなふうに過去を診断する試みに賛成しない。「彼がそんな病気を持っていたとは思いもしなかったよ」と彼は言う。「ずいぶん社会性がなかったけど、当時、ぼくたちみんながそうだったんだしね。」

これに対して、リップマンはその可能性を受け入れている。彼女は推測の材料を提供する息子の幼児期の話をいくつか思い出した。自閉症の顕著な兆候の 一つは不快な音や色に対する過敏さだが、リップマンはこの点で二つの逸話を思い出した。「リチャードが幼児だったときに、私達は彼をビーチに連れて行きま した」と彼女は言う。「波打ち際に行く二ブロックか三ブロック手前で彼は大きな声で泣き始めました。私達は三回目まで、いったいどうなっているのかわかり ませんでした。波の音がいやだったんです。」彼女は同じく泣き始める反応を色についても思い出す。「私の母の髪の色が明るい赤だったんですが、抱こうとしてしゃがむたびに泣かれたんです。」

最近、自閉症について書いた本を読むようになって、そんなエピソードが偶然の一致以上のものだったと思うようになったとリップマンは言う。「リ チャードには、自閉症の子供の資質がいくらかあったと感じています。」彼女は言う。「そのとき、自閉症についての知識がほとんどなかったことが残念で す。」

しかしいつしか、ひとりでに息子は順応するようになったとリップマンは言う。7 歳までに、都市の地下の迷宮のような路線図を描きながら覚え、地下鉄の列車の前窓に立つのが好きになっていたと彼女は言う。それは、列車に乗るたびに大き な騒音に順応する能力が必要な趣味だった。「彼を悩ますのは最初の騒音だけでした」と、リップマンは言う。「彼はその音がショックみたいでしたけど、順応 する方法を神経が学んだんですね。」

たいていのこと、普通の少年なら持っている興奮や、エネルギーや、社会適応の技術を息子は示していたとリップマンは回想する。それはストールマンの 家族を打ちのめした一連のトラウマ的出来事の後は違ってしまったというのではない、と彼女は言う。その出来事の後は、息子は内向的になり、感情的に遠くに 行ってしまったのではあるが。

最初のトラウマな出来事は、アリスと、リチャードの父、ダニエル・ストールマンの離婚だった。彼女と前夫は、息子の精神的打撃に対して心の準備をさ せようと試みたが、その打撃は壊滅的だったと彼女は言う。「何が起こっているかを私達が最初に彼に教えたとき、彼は聞き流していました。」リップマンは回 想する。「それでも、新しいアパートメントに引っ越したときに、その現実が彼の顔をたたきました。最初に言ったのは、『パパの家具はどこ？』でした。」

次の 10 年間、ストールマンは、平日は母とマンハッタンのアパートメントで過ごし、週末はクイーンズの父の家で過ごした。行ったり来りの往復はストールマンに一組 の対照的な育児スタイルを学ぶ機会を与えたが、ストールマンは、今日に至るまで、自分が子育てするというアイデアを固く拒絶するようになった。ストールマ ンは、父のことを語るときに尊敬と怒りのバランスをとろうとする。父は第二次世界大戦の古参兵で 2001 年初頭にこの世を去った。一方では、時が来たときにはもっと連合国の力になれるようにフランス語をストールマンに学ぶよう精神的に後押しした人物であり、 もう一方では、いつも思いやりの全くないやり方でこき下ろす方法を知っていた親だった。7

「父は、かなり怒りっぽくて」とストールマンは言う。「怒鳴りはしないけれど、いつも、冷たい、人をめちゃめちゃにするようなやり方で批判する方法を見つけたんです。」

母親のアパートメントでの生活に関していえば、ストールマンはそれほど両義的ではなかった。「それは戦争でした」と彼は言う。「『おうちへ帰りたい。』とみじめに言っていました。 自分は持てない存在しない場所を言っていたのです。」

離婚後の最初の数年間は、父方の祖父母の家でストールマンは平穏さに包まれることができた。8 歳のときに、そして 10 歳のときに、祖父母が相次いで亡くなった。ストールマンにとって、その喪失は打撃だった。「ぼくはよく遊びに行って、愛されてると感じていました。やさし い環境の中にいたのです。」ストールマンは回想する。「カレッジに行くまで、ぼくが見つけた唯一の居場所でした。」

リップマンは、リチャードの父方の祖父母の死を、二番目のトラウマな出来事として挙げる。「彼は本当に動揺していました。」彼女は言う。「祖父母と はとても仲良くしていました。亡くなる前は、彼はすごく外交的でした。ほとんどガキ大将タイプ。亡くなった後は、感情的にもっと内向的になって行きまし た。」

ストールマンから見れば、感情的に内向的なのは思春期の苦痛を何とかしようという試みにすぎなかった。「純粋な恐れ」というレッテルを自分の 10 代に貼り、ストールマンは、ときおり音楽を聴く人でざわざわする人ごみの真中にいる耳が聞こえない人間であるように感じていたと言う。

彼の追放感を回想しながら、ストールマンは言う。「他人が何を言っているのか、ぼくには理解できないだろうと思うことがよくありました。」「言葉を 理解することはできても、会話の下では何かが進んでいて、それがわからない。他人が話題にしてることが何で面白いのかわからない。」

それがつくりだしたすべての苦悩のため、思春期はストールマンの個性に関する意識を強める効果を持った。クラスメートのほとんどが髪を伸ばすように なった頃、ストールマンは短くしているのが好きだった。10 代の世界全体がロックンロールを聴いていた頃、ストールマンはクラシックを好んだ。熱心な SF ファン、マッド・マガジン、テレビの深夜番組。ストールマンは、親からも仲間からも理解されなかったが、極めて異色のパーソナリティを養っていった。

「ああ、駄洒落ね。」息子の 10 代の記憶を思い出して、リップマンは今でもいらだって言う。「夕食の席で、彼が駄洒落として応じられないようなことは、言ってはいけないんです。」

家の外では、彼の天賦の才能に気を配ってくれる大人に対して、ストールマンはジョークを慎んでいた。最初の一人は 8 歳か 9 歳のときのサマー・キャンプの指導員で、ストールマンに、IBM 7094 コンピュータの印刷されたマニュアルを手渡した。数と科学に魅せられていた 10 代手前の子供にとって、そのプレゼントは神様の贈り物だった。8 まもなくストールマンは、7094 の取扱説明書の紙にプログラムを書き上げた。実行するコンピュータは周りになかった。彼はそのために使う本物のアプリケーションを持っていなかったが、どうしてもプログラムを書きたかった - プログラムなら何でも良かった。彼はキャンプの指導員に、何でも良いからコードを書くのに大切なことを言ってくれるよう求めた。

最初のパーソナル･コンピュータができるのはまだ十年先のことで、ストールマンが初めてコンピュータにアクセスできるようになるまで、それから数年 待たざるを得なかった。高校の最終学年のとき、ついに最初のチャンスが訪れた。 マンハッタンのダウンタウンにあったもう現存しない研究施設、IBM ニューヨーク科学センターが、ストールマンに最初の本物のプログラムを書く試みの機会を提供した。彼の夢は、PL/I プログラミング言語のプリプロセッサを書くことだった。言語の機能の一つとしてテンソル代数の加法規則を追加する構想だった。「ぼくが最初に書いたのは、 PL/I でした。コンパイルした PL/I プログラムでは大きすぎてそのコンピュータには合わなくなって、その後アセンブラ言語で書き直しました。」と彼は回想している。

高校を卒業した年の夏の間、IBM ニューヨーク科学センターが彼を雇った。Fortran（フォートラン）の数値解析プログラムを書くのが任務だったが、彼は二三週間でそれを書き終えた。 二度とそれで書くことを承知しなかったほど嫌いな Fortran 言語で書くことを求められたためだった。そして、夏の残りは APL のテキストエディタを書いて過ごした。

同時に、ストールマンは、ロックフェラー大学生物学部で研究室助手の地位を得ていた。彼は、すでに数学か物理のキャリアに向かっていたのだが、ス トールマンの分析的思索力は所長に強い印象を与えた。それはストールマンがカレッジに出発した数年後に予期せぬ電話をリップマンが受け取るほどのものだっ た。「ロックフェラーの教授でした。」とリップマンは言う。「リチャードがどうしているか知りたがっていました。彼はコンピュータをやっていると知って驚 いていました。教授はリチャードに生物学者として自分よりも大きな未来があるとずっと思っていたそうです。」

ストールマンの分析力はコロンビアの教授達にも同じく深い印象を与えた。ストールマン自身が彼らの憤慨の対象になっているときでさえそうである。 「講義をしていると、いつも 1 時間に一つか二つ間違いを見つけるんです。」とブライドバートは言う。「そして恥ずかしがらずに、すぐ教授にそれを教えるんですよ。とても尊敬されたけ ど、人気はあまり得られなかった。」

ブライドバートの語る逸話は、ストールマンの苦笑いを引き出す。「ぼくは、ちょっとむかつく奴だったかもしれない」と彼は認める。「けど、教師たち の中に自分と同類の精神を見つけていたんです。彼らも学ぶのが好きですからね。子供たちはたいていそうじゃない。少なくとも、同じような意味では好きじゃ ない。」

しかし、進んでいる子供達と一緒の時間を土曜日に過ごすことで、ストールマンは、社会性をより豊かにすることのメリットを考えるようになった。大学 まで早く近づいたために、コロンビア・サイエンス・オナーズ・プログラムの多くの生徒と同じように、ストールマンは、彼の望む学校を 2 つの選択肢にまで絞った。つまり、ハーヴァードか MIT だ。彼女の息子がアイビー・リーグに入りたいというのを聞いて、リップマンは懸念した。15 歳の高校生として、ストールマンは、いまだに教師や管理者とけんかしていた。1 年前でさえ、アメリカ史、化学、フランス語、幾何学では、A をとっていた。でも、作文の宿題をボイコットして、英語は F だった。そういう間違いは MIT ではわけ知りのくすくす笑いをされるだけかもしれないが、ハーヴァードでは赤信号になる。

彼女の息子が 2 年生のとき、リップマンは、セラピストのアポイントメントをとった。セラピストはすぐにストールマンが論文を書きたがらないことや、教師陣とうまくやりたがらないことについて懸念を述べた。彼女の息子は、確かにハーヴァードに行くぐらいの知力があった。期末のレポートを要求される大学の授業をやりとおせる 忍耐力があるのだろうか。セラピストは、試してみるように提案した。もし、ストールマンが期末レポートを要求される英語の授業も含むニューヨークの公立学 校で 1 年間をやりとおせるなら、ハーヴァードでもそれができるでしょう。ストールマンは 2 年生を終えるとすぐ、ダウンタウンの公立高校のサマースクールに登録した。そして高校に入った頃は避けていた必修の人文科目のクラスの遅れを取り戻し始め た。

秋までに、ストールマンは 84 番ストリートにあるルイス D.ブランダイス高校で、ニューヨーク市の高校生の主流の中に戻った。コロンビア大学での土曜日の勉強に比べれば、授業中ずっと座っているのは簡単ではな かった。しかし、リップマンは、彼のスタートラインに立つ能力を誇りを持って思い出す。

「彼はある程度ぺこぺこしなくちゃいけなかったけど、そうしたんです。」とリップマンは言う。「私が呼び出されたのは一度だけ。ちょっと奇跡ね。リ チャードに授業を邪魔されているという微積分の先生の話でした。彼は何をしてるんですか、と聞いたら、リチャードはいつも、先生は間違った証明を使ってい ると非難していたんです。『で、彼は正しいんですか』って聞きました。その先生が言うには『そうなんです。でも、ぼくはクラスで言うことはできないんで す。理解できないだろうから。』ということでした。」

ブランダイスでの最初の学期が終るまでに、収まるべきところに収まっていった。英語の評点で 96 点を得て 2 年前に 60 点だった不名誉をぬぐい去った。よい尺度としては、ストールマンは、アメリカ史と AP 微積分と微生物学でトップになった。最高は物理の 100 点満点。社会性はなかったが、ストールマンは 789 人中 4 位でブランダイスの 11 ヶ月

を終えた。

教室の外では、平日は研究室助手の任務をこなすためにロックフェラー大学へ急行し、コロンビアの土曜学校へ行くときにはベトナムの抗議運動をしてい る人達を避けながら、なおさら熱心にストールマンは自分の研究を追求した。そんな頃、サイエンス・オナーズ・プログラムの生徒達がカレッジの選択を話題に していた、クラスの始まる前の雑談に、ストールマンがついに加わったことがあった。

ブライドバートは回想する。「ほとんどの生徒は、ハーヴァードか MIT に行こうとしていました。もちろん、アイビー・リーグの他の大学に進む人も少しはいましたけどね。話がクラスを一回りしたときに、明らかにリチャードがま だ何も言ってないことがわかった。誰だったかわからないけれども、誰かが勇気を出して、どうするつもりって聞いたんです。」

30 年後、その瞬間をブライドバートは鮮明に思い出す。ストールマンもこの秋にハーヴァードに入学するというニュースを明かしたとたん、教室の空気が固まった。それがきっかけとなったように、ゆっくり彼の口が開き、それは会心の微笑みに変わった。

ブライドバートは言う。「言外に言ってたんだ。『そう。ぼくをまだ追い出せないね』って。」

後注

1 この章の背景の主な出典は、マイケル・グロス（Michael Gross）によるインタヴューの『リチャード・ストールマン。高校の不適応者、自由ソフトウェアのシンボル、マッカーサー奨学金の天才。』である。マイ ケル・グロスは、いわゆる「ベビー・ブーム」世代の注目すべき人物に関する 1999 年のインタビューを集めた本、"Talking About My Generation" の著者である。ストールマンはこの本に入っていなかったが、Gross は、その本のウェブサイトでオンライン版の追補をした。私が見つけてから、その URL は何度か変わっている。それを探していた読者によると、そのインタビューはここで見つかる。
http://www.mgross.com/MoreThgsChng/interviews/stallman1.html

2 RMS: 私はこういう話をしていたことを憶えていない。言えるのは、今ではそれらの見解に強い異論があるということだけである。私は十代の頃に、大半の人が人生で 遭遇する困窮に深い同情を欠いていた。私の問題は別のところにあった。あらゆるレベルで私たちがそれを止めるように組織しない限り、富者がいかにして大半 の人々を貧困に陥れるかを私は十分に理解していなかった。多くの人々にとって、たとえば貯金をしないでお金を全部使ってしまうような、愚かな事をしてしま う社会的圧力に抵抗するのがいかに難しいことか私は理解していなかった。なにしろ、私は自分に対する圧力でさえほとんど気付かなかったのだ。さらに、60 年代の労働組合はその頃非常に強力だったが、しばしば傲慢で腐敗していた。だが今日、組合はとても弱くなった。その結果、経済成長のときでも、もっぱら経 済成長の利益は富者に与えられている。

3 前掲書。

4 次を参照。Judy Steed, Toronto Star, BUSINESS, (October 9, 2000): C03

自由ソフトウェアと社会的協力の彼のビジョンは、彼の私生活の孤独な性質と対照的 だ。

グレン・グールドのように風変わり、カナダのピアニストは、同じように、輝かしく、歯切れ良く、孤独だった。ストールマン自身も、自閉症に苦しんでい るのだと、ある程度考えていた。自閉症とは、彼が言う、ひとと付き合うことが難しいという状況である。

5 次を参照。Steve Silberman, "The Geek Syndrome," Wired (December, 2001), http://www.wired.com/wired/archive/9.12/aspergers_pr.html.

6 次を参照。John Ratey and Catherine Johnson, "Shadow Syndromes."

7 残念ながら、本書のためにダニエル・ストールマンにインタビューする機会は得られなかった。本書の調査の最初の頃、ストールマンから、父がアルツハイマー にかかっていると知らされていた。2001 年後半に調査を再開して、悲しいことに、その年の初めにダニエル・ストールマンが亡くなっていたことを知った。

8 無神論者のストールマンは、この記述におそらく難癖をつけるだろう。そこで、この記述はストールマンに歓迎できない何かがあるのだと言えば満足してもらえ よう。注 1 の Gross (1999)を参照されたい。「ぼくはコンピュータのことを耳にすると、それを見たくなって、そして遊んでみたくなった。」

第四章　神を弾劾せよ

二人の関係は緊張をはらんでいたが、リチャード・ストールマンは母から注目すべき特徴を一つ受け継いでいた。進歩的な政治への情熱である。

だが、その遺伝形質が現われるには数十年を要した。彼の人生の最初の数年間はストールマンが現在認めているような「政治的真空」の中にあった。1 アイゼンハウアー時代の多くのアメリカ人と同じように、ストールマンの一家は 40 年代の戦争中に失われた正常な状態を取り戻そうとして、50 年代を生きていた。

「リチャードの父親も私も民主党員だったけど､取り立てて何もしなくても十分幸せだったの､」とリップマンはクイーンズでの家族時代を思い出して言う。「地域にも国の政治にもあまり関わろうとはしなかったわ。」

しかし、1950 年代の終りにアリスがダニエル・ストールマンと離婚してからすべてが変りはじめた。マンハッタンへ戻ることは、住所の変更以上のことを意味していた。それは新たに独立した存在になり、平穏さが軋みながら失われていくことを意味していた。

「私の政治的積極主義のきっかけは、クイーンズ公立図書館に行ったときに離婚の本が蔵書にたった一冊しかないことを発見したときだったと思いま す。」とリップマンは回想する。「少なくとも私の住むエルムハーストでは、離婚はカトリック教会が厳しく制限していました。私達の生活への無言のコント ロールに初めて気づき始めたんです。」

子供の頃に住んでいたマンハッタンのアッパーウエウストサイドに戻ると、15 年前、彼女がハンター大学進学のために転居して以降、そこが受けてきた変化にリップマンはショックを受けた。戦後、住宅需要が急上昇したことで、このあた りは政治の戦場と化していた。一方の側に、市内に引越してくるホワイトカラー労働者の増加に応じて、地区の多くのブロックを建て直すことを望む開発派の市 議会議員とビジネスマンがいた。もう一方の側には、この地区に自分達でも支払っていける安息の地を見い出した貧しいアイルランド人とプエルトリコ人の借家 人がいた。

最初、リップマンはどちら側を選ぶべきか分からなかった。新参の住人として、彼女は新規の住宅供給の必要性を感じていた。だが、最低限の収入しかな いシングルマザーとして彼女は、主に裕福な居住者に提供するための開発プロジェクトの増大に対して、貧しい賃借人たちと不安を共有していた。義憤を感じた リップマンは、彼女の地区をアッパーイーストサイドのクローンにしようと企てる支配集団と一戦交える方法を探し始めた。

地域の民主党本部を最初に訪ねたのは 1958 年だった、とリップマンは言う。働いている間、息子の世話をしてくれる託児所を探していて、低収入の住民に提供されている、ある市営託児所の環境に彼女は 唖然とした。「思い出せるのは、腐ったミルクの臭いと、暗い廊下と、支給品不足だけね。私は私立保育園の先生だったの。その落差はとても大きかったわ。私 達はその部屋を一目見るなり帰りました。それが私を奮起させたんです。」

だが、党本部への訪問は期待外れだった。党本部を「かの有名な煙もうもうたる部屋」と

形容するリップマンは、貧しい住民に対する市当局の露骨な敵意 の背後に、実は党の腐敗があることに初めて気づいたと言う。リップマンは本部に戻らず、民主党を改革し、タマニー・ホールの支配集団の最後の痕跡を一掃す ることを目指す多くのクラブの一つに加わることを決めた。ウッドロー・ウィルソン / FDR 民主党改革クラブと名乗ったリップマンと彼女のクラブは企画立案会議や市議会に現れて、強い発言権を要求し始めた。

「私達の第一目標は、タマニーホールのカルミネ・デサピオとその腹心と戦うことでした、」2 とリップマンは言う。「私は市議会に対する私達の代表をしていて、この地区にもっと贅沢な住宅を供給するだけじゃなく、実のある都市再生計画を立案することにしゃにむに取り組んでいました。」

そのような取り組みは 1960 年代に、さらに大きな政治活動へ開花していくことになる。1965 年頃には、リップマンは、ウィリアム・フィッツ・ライアンのような候補に「何でも言える」支持者になっていた。ウィリアム・フィッツ・ライアンは、改革クラブの支持を得て下院に選出された民主党員で、ベトナム戦争に反対した最初のアメリカ下院議員の一人だった。

ほどなく、リップマンもアメリカのインドシナへの介入に反対して声をあげた。「ケネディーが派兵したときから、ベトナム戦争に反対でした。」彼女は言う。「レポーターやジャーナリストがこの紛争の初期段階を取材した記事を読んでいました。 この紛争は泥沼になるという彼らの予想を確信しました。」

こうした反対意見は、ストールマンとリップマンの一家に浸透していた。 1967 年にリップマンは再婚した。 彼女の新しい夫、モーリス・リップマンは州空軍の少佐で、 戦争に対する反対を表明するために将校の地位を辞した。 リップマンの再婚相手の息子アンドリュー・リップマンは MIT にいて、 学生として一時的に兵役を猶予されていた。 しかし、実際そうなったように、そんな猶予が消し飛ぶような兵士募集への兆しがあり、 アメリカによる戦争拡大が間近に迫っていた。 最後に、アンドリューより年少とはいえ、リチャードも、 戦争が 1970 年代まで長引けば、ベトナムかカナダかの選択を迫られることになろうとしていた。

「ベトナムはわが家の大問題でした。」とリップマンは言う。「いつもその話をしていました。戦争が長引いたらどうするのか、 徴兵されたら、リチャードやアンドリューはどうすべきか。 私達は全員が戦争や徴兵に反対でした。 この戦争や徴兵はまったく人の道を外れたたものだと思っていました。」

ストールマンにとっては、ベトナム戦争は複雑な感情、 当惑、恐怖、そして最終的に、心底からの政治的無力感をひき起こした。 私立学校の温和で権威主義的な宇宙の中でかろうじてやっていける子供だった ストールマンは、軍隊の新兵訓練所の存在を考えるだけで身震いした。彼はそれをやり通して正気で帰って来れると思わなかった。

「ぼくは不安に打ちのめされて、何をすべきかを想像することもできず、 デモに行く度胸もありませんでした。」 1971 年、連邦政府がついに学生の徴兵延期を取りやめる決定を

下したとき、 3 月 18 日の誕生日に徴兵の抽選で、恐れていた低い数字を引き当てたストールマンは、そう回想する。 「カナダやスウェーデンに移住するなんてことは、思いもよりせんでした。 自分で立ち上がって、どこかに行くというような考えはね。 どうすれば、ぼくにできたでしょうね。自活していく方法なんて知りませんでした。 物事に自信をもってアプローチしていくようなタイプじゃなかったんです。」

はっきり意見を言う家族のメンバーに感心すると同時に自分が恥ずかしかった、とストールマンは言う。 ソンミ村の虐殺を第二次大戦のナチの残虐行為になぞらえた父の車のバンパーのステッカーを思い出しながら、父の怒りの仕草に「興奮した」と言う。 「そうする父を偉いと思った」とストールマンは言う。 「でも、ぼくに何かができるとは想像できませんでした。 ぼくは徴兵という大きな力がぼくを破壊してしまうことを恐れていました。」

しかし、結局、彼が反戦運動への関心を失ったのは、その論調と方向性のためだ、とストールマンは言う。 サイエンス・オナーズ・プログラムの他のメンバーと違わず、 彼はコロンビア大学の週末のデモをちょっと気になる光景ぐらいに見ていた。3 結局、反戦運動を動かしていた非合理的な力は、 他の若者文化を動かしている非合理的な力と区別できなくなった、とストールマンは言う。 ビートルズ崇拝のかわりに、ストールマンの年代の少女たちのグループは、 突然、アビー・ホフマンやジェリー・ルービンのような扇動家を崇拝するようになった。 すでに十代の自分の同輩達を理解するために苦闘していた一少年にとって、「メイク・ラブ、ノット・ウオー(make love not war;戦争を止めてセックスしよう)」のようなスローガンには、からかわれているような気がした。ストールマンは、少なくとも東南アジアでの戦争をメイ クすること望んでいなかったが、彼をメイク・ラブに誘う者もいなかった。

「カウンター・カルチャーはあまり好きじゃなかったですね、」とストールマンは認める。「ああいう音楽は好きじゃなかったし、ドラッグも好きじゃありませんでした。 ドラッグは怖かったんです。特に反知性主義はいやで、テクノロジーに対する偏見もいやでした。 結局、コンピュータが好きだったんです。 そして、しばしば出くわす軽率な反米主義も嫌でした。 ベトナムでのアメリカの振る舞いを容認できないなら 北ベトナムを支持すべきだとひどく短絡的に考える人達がいました。 彼らはもっと複雑な立場を想像できなかったんだと思います。」

こういうコメントは臆病な気持ちを和らげるものだが、 ストールマン自身の政治的成熟への鍵となった特性も際だたせている。 ストールマンにとって、政治的信条は個人的信条に直接連動していた。 1970 年まで、数学と科学の領域以外でストールマンが信頼しているものはほとんどなかった。 それでも、数学への信頼は、反戦運動を純論理的なことばで検討する十分な足場を与えてくれた。 そうする過程で、ストールマンは論理性の欠如に気づいた。 ベトナム戦争には反対だったが、ストールマンには 自由を擁護したり、不正を正すための手段としての戦争まで否定する理由はないと思われた。

1980 年代に、もっと自信を持ったストールマンは、ワシントン D.C.での堕胎する権利の大衆集会に参加することで、過去の不活動を埋め合わせることにした。「ベトナム戦争に抗議する自分の義務にしくじった以前の自分に満足できなくなったんです。」と彼は説明する。

1970 年、ストールマンは政治やベトナム戦争をめぐる夜毎の夕食時の会話を後にして、ハーヴァードへと旅発った。 振り返って、母親のマンハッタンのアパートからケンブリッジの寮への変遷を、 ストールマンは「逃亡」と表現している。 だが、この変遷したストールマンを見ていた仲間たちには、 およそ彼が解放感を味わっているようには見えなかった。

「ハーヴァードでの最初の頃、彼はかなり悲惨に見えました、」と、 サイエンス･オナーズ･プログラムのクラスメートでやはりハーヴァードに進学していたダン・チェスは回想する。 「彼は人との交際が本当に苦手だったんだろうと思いますが、 ハーヴァードではそれは避けられなかったんです。ハーヴァードはとことん社交的なところでした。」

この変遷を楽にするために、ストールマンは数学と科学という自分の得意分野に退却した。 サイエンス･オナーズ･プログラムの大抵のメンバーと同様、 ストールマンもまた、ハーヴァードの新入生用の数学「強制収容所」として伝説的な「新兵訓練所」クラス、Math 55 の資格試験に難なく合格した。 そのクラスで、サイエンス･オナーズ･プログラムのメンバーは固く結束していた。「ぼくらは数学マフィアでした。」とチェスは笑って言う。「SHP (サイエンス･オナーズ･プログラム) に比べれば、 ハーヴァードなんてたいしたことありませんでした。」

だが、自慢していられるためには、ストールマンやチェスを始めとする SHP 出身者たちは、 Math 55 に及第しなければならなかった。 二学期間で四年分の数学が学べることを約束するこのコースは、真に献身的な者たちだけに好意的だった。 「驚くべきクラスでした、」とかつての「数学マフィア」のメンバーで、現在は ペンシルベニア大学の数学教授である、デビッド・ハーバター（David Harbater）は言う。 「間違いなく、入りたての学生にこれほど集中的に、これほど進んだことを教えたクラスは かつて無かったと言えます。 それを分かってもらうには、ぼくたちは、第二学期には、 バナッハ多様体の微分幾何について論じ合っていた、というのが一番伝わるでしょう。 大抵これでみんなびっくりします。 というのは、 ふつうバナッハ多様体のことは、大学院の二年生になって始めて口にするからです。」

始めは 75 人の学生がいたが、クラスはたちまち溶解して、2 学期の終りには 20 人になっていた。 この 20 人の中で、とハーバターは言う､「自分が何をしているのかが分かっていたのは 10 人だけでした。」 この 10 人の中で、8 人は後に数学教授に、1 人が物理を教えるようになりました。

「もう一人が、」とハーバターは語気を強める､「リチャード・ストールマンでした。」

Math 55 のクラスメートだったセス・ブライドバートは、そのころからストールマンが仲間より抜きん出ていたことを覚えている。

「彼はうるさ型だったんですが、それはとても変わった風にでしてね、」とブライドバートは言う。「数学の標準的な手法なんですが、誰もが間違う箇所があるんです。 それは何かの関数を定義するときに記法を誤用することで、 関数を定義してから、それがきちんと定義されていることを証明する、というものです。 最初は彼もそういうプレゼンをしたんですが、 それを除けば、彼はある関係を定義して、それからそれが関数であることを証明したんです。 それは証明としては全く同じなんですが、彼は正しい用語を使いました。 これが他の人にはできなかったんです。 彼はこんな風でしたね。」

Math 55 でリチャード・ストールマンは、輝かしい評判を培い始めた。 ブライドバートはうなづくが、負けじとばかりにチェスは、ストールマンが クラスで最高の数学者であることを実証したのは、その翌年だという。「それは実解析のクラスでのことです、」と、現在ハンター大学の数学教授であるチェスは言う。 「私はまざまざと思い出します。複素数値の測度に関するある証明で、 リチャードは基本的に変分法からからヒントを得たアイディアを持って現れました。 誰かがある問題を、独自な仕方で鮮やかに解決するのを目の当たりにしたは、これが初めてでした。」

チェスにとって、それは悩ましい瞬間でもあっった。 透明なガラス窓に飛び込む小鳥のように、 洞察力にも到達しえないレベルがあることに気づくのに一定の時間を要した。

「数学の世界では、それが事実なんです、」とチェスは言う。「一流の数学的才能を認めるのに、一流の数学者である必要はありません。 私はその域には達しているとはいえますが、自分が一流の数学者だとはいえません。 リチャードが数学者になることを選んでいたら、彼は一流の数学者になっていたでしょう。」4

ストールマンは教室ではうまくやっていたが、 社交面では同じくらいうまく行かなかった。数学マフィアの他のメンバーが集って、Math 55 の問題集に取り組むようなときでも、ストールマンは一人で事に当たることを好んだ。 日常生活でも同様だった。 ハーヴァードでの居住条件についても、ストールマンは自分の好みを明言した。「ぼくは、見えないし、聞こえないし、触れもしないルームメートがいい、と言ったんです、」と彼は言う。 ハーヴァードのお役所的慎重さには稀な素早さで、ハーヴァードの住宅斡旋課はこの要求を受け入れ、 新入生のときからストールマンに一人部屋を与えた。

新入生のときにストールマンと同じ寮に入った唯一の数学マフィアのメンバーであるブライドバートが言うには、ストールマンはゆっくりと、だが確実に他の学生とのつき合い方を覚えていった。 ストールマンの論理的な洞察力に感心した他の寮生たちが、 夕食クラブや寮の食堂で知的な議論が始まると、彼の発言を歓迎したことを、彼は思い出す。

「ぼくたちは、世界の諸問題の解決やあることの結果がどうなるかということについて、いつものように角突き合わせていました、」とブライドバートは回想する。「例えば、誰かが死なない薬を発見したとする。君ならどうするだろう。 その政治的な帰結はなんだろう。それをみんなに与えたら、世界は人口過剰になり、みんなが死んでしまうだろう。 それを制限して、今生きている人はもらえるが、その子どもたちはもらえないようにしたら、 そ

れを手に入れられない最下層の人々が生まれるだろう。 リチャードは、ある決定に伴う見通しにくい状況を、誰よりもよく見通すことができました。」

ストールマンはその議論をまざまざと憶えていた。「ぼくはいつでも不死の味方でした、」と彼はいう。「死んでしまったら、今から 200 年後の世界は見られないじゃないですか。」好奇心をそそられた彼は、色々な知人に、もしも不死を提供されたら望むかどうか尋ね始めた。「多くの人が、不死を悪いことと考えているのにはびっくりしま した。」老いぼれた人生を生きても役に立たないから死は良い事なのだ、そして加齢は人々に死の準備をさせるからから良い事だ、と多くの人が、組み合わせの 循環性を評価せずに言っていました。

ストールマンは一流の数学者で、一流の論客だと認められていたが、 自らの秀才の誉れが評価されることになるような露骨な競争イベントを避けた。 ブライドバートの回想によれば、ハーヴァードの初年度の終りに、 ストールマンは臆面もなくパトナム試験から逃げた。 この試験は、全米とカナダの数学の学生を対象とした有名な試験で、 学生たちに、同僚たちとの比較での自分の知識を測る機会を与えるだけでなく、 大学の数学科に人材を集めるための主要なツールでもあった。 キャンパス伝説によれば、最高得点者は、ハーヴァードを含む好きな大学の大学院に進学可能と自動的に評価された。

Math 55 と同様、パトナムは、苛酷だが評価の高い試験だ。 2 部からなる 6 時間の試験で、明らかにモミ殻とモミをより分けられるように設計されている。 サイエンス・オナーズ・プログラムと Math 55 の経験者であるブライドバートは、ためらうことなく、 それが自分が受けた中で一番難しいテストだったと言う。 「その難しさがどんなものかというと、」とブライドバートは言う、 「最高点は 120 点で、初年度のぼくの成績は 30 点台でした。 それでも、この成績は全国で 101 番になれるくらいには良かったんです。」

ブライドバートによれば、驚いたことに、クラスで 1 番のストールマンが、 このテストを受けていなかった。 そこで彼とクラスメートたちは食堂でストールマンを問詰め、説明を求めた。 「彼は失敗するのがこわかったんだ、と言っていました、」とブライドバートは回想する。

ブライドバートと彼の友人はすぐに覚えていた問題をいくつかを書き出して、 ストールマンにやらせてみた。「彼は全部解けました、」とブライドバートは言う、 「だから、彼の言う失敗とは、一番になれないとか、何かを誤解してしまう、 と言う意味だと思いました。」

ストールマンはこのエピソードを幾分違って覚えている。 「ぼくが覚えているのは、彼らがぼくにいくつか問題をくれて、 多分その一つを解いたとは思いますが、全部解いたなんてことはまずありません、」と彼は言う。 だが、ストールマンは、恐れがテストを受けなかった第一の理由だった、というブライドバートの 回想には同意する。 教室でクラスメートや教授たちの知的な弱点を指摘することはためらわなかったが、ストールマンは角突き合わせての競争という観念を嫌い、危惧した - だから近づかなければ良い。

「ぼくがチェスを好きになれない理由も同じです、」とストールマンは言う。 「チェスをしようとすると、一つでもミスを犯すことへの恐怖に囚われてしまって、 そのためにゲー

ムの始めの方で、愚かなミスを連発してしまうんです。 恐れが予感を実現させてしまうわけです。」彼はチェスをやらないことで問題を避けた。

こうした恐怖心がストールマンが数学を専攻するのを避けた究極の原因だとするのには、議論の余地がある。 ハーヴァードの一年目の終り頃には、ストールマンを数学から引き離して行くような、別の関心領域が生じていた。 高校時代からくすぶり続けていたコンピュータ・プログラミングへの情熱が、 一人前に成熟しつつあった。 数学科の学生が時おり芸術や歴史の授業に避難所を求めるときに、 ストールマンは、コンピュータ科学研究室にそれを求めた。

IBM のニューヨーク科学センターで本物のコンピュータ・プログラミングを始めて味わったことが、 ストールマンにもっと知りたい、という気持を呼び起こした。 「ハーヴァードの一年目の終り頃に、思い切って、コンピュータ関係の研究室を訪ねて、 そこに何があるのか知ろうと思いました。 そして、ぼくが読んでいいようなマニュアルの余分なコピーがないかときいてみました。」マニュアルを自室に持ち帰って、それを自分が既に知っている他のマシンのものと比較し、ストールマンはマシンの仕様を調べた。

ハーヴァード 1 年目の終りに近いある日、ストールマンは MIT の特別な研究室のことを耳にした。 その研究室は、キャンパス外のテック・スクウェアのビルの 9 階にあり、 そこの新品の設備で高度な研究が行われることになっていた。 噂によれば、この研究室自体は、人工知能という最先端の科学を研究するところで、 それに見合う最先端のマシンとソフトウェア・プログラムを誇っていた。

この話に引き付けられて、ストールマンはそこに行ってみることにした。

その道のりは短く、徒歩で 2 マイル、電車で 10 分にすぎなかったが、ストールマンはすぐに、MIT はハーヴァードと同じ惑星の反対側の極にあるような感じがすることに気づいた。ハーヴァードの広々としたコロニアル風の「陽」に対して、MIT のキャンパ スが、継りあったオフィスビルが迷路のようにもつれあっている様は、美的な「陰」を示していた。同じことは主体となる学生にも言えた。かつて高校に適応で きなかったおたくの集まりであり、政治的に力のある卒業生よりも、いたずらをこよなく愛することで通っていた。

この陰・陽の関係は、AI ラボにも及んでいた。ハーヴァードのコンピュータ研究室と異なり、大学院生の門番はおらず、掲示板に端末待ちのリストはなく、「見るだけで、触わらないこ と」といった露骨な雰囲気もない。そのかわりに、ストールマンが見たのは、開放された端末や、何かの AI 実験の産物と思われるロボット・アームの集まりだった。ラボの職員に出会ったときに、彼は、ラボには知りたがりの学生に貸し出せるような予備のマニュアル がないか聞いてみた。「彼らはいくつかは持っていましたが、多くは文書化されていませんでした、」とストールマンは回想する。最後のものを文書化せずに新 プロジェクトをどんどん進めるハッカーの傾向に言及しながら、苦笑して彼は付け加えた。「つまり、彼らはハッカーだったんです。」

ストールマンはマニュアルよりましなものをもらって帰った。仕事だ。彼の最初のプロジ

ェクトは、PDP-10 で走らせる PDP-11 シミュレータを書くことだった。次の週に AI ラボに戻ったとき、空いている端末を拝借して、ソフトウェアのコードを書き始めた。

今にして思えば、AI ラボが一見の身元の知れない部外者を快く受付けたことは別に不思議ではない、とストールマンは見る。「その頃そうだったし、」とストールマンは言う。「今でもそういうやり方なんです。誰かに会って、いいと思えば雇います。なぜ待つんです？何にでも堅苦しく、官僚主義を適用しようとする人たちは、的をはずしています。 もしいい人がいたら、その人は細々とした長い採用プロセスを経ることはないんです。 その人はコピュータの前に座って、コードを書いているべきなんです。」

「官僚的で堅苦しい」感じを味わうには、ストールマンはハーヴァードのコンピュータ研究室に行けばよかった。そこでは、端末へのアクセスは大学内で の序列に応じて配分されていた。学部生だったストールマンはいつも申請登録するか、深夜まで待たねばならなかった。その頃になると、大方の教授や大学院生 が日課を終えるのだ。待つことは、難しくはなかったが、いらいらした。鍵のかかった教授たちの部屋には半ダースもの使える端末が空いているのを知りなが ら、共有端末を待つというのは、不合理の極みに思えた。 ストールマンはたまにハーヴァードのコンピュータ研究室を訪れることもあったが、AI ラボのより平等主義的な方針が好きだった。「一陣のさわやかな風みたいでした、」と彼は言う。「AI ラボではみんな地位より仕事に関心をもっているようでした。」

ストールマンはすぐに、AI ラボの早いもの勝ちポリシーは、炯眼な数人の努力に負うところが大きいことを知った。その多くはプロジェクト MAC 時代からの残留者だった。プロジェクト MAC は最初の時分割オペレーティングシステムを誕生させた、国防省出資の研究プログラムだ。そのうちの何人かは、すでにコンピュータの世界で伝説になってい た。ラボお抱えの Lisp エキスパートであり、MacHack? の作者であるリチャード・グリーンブラットがいた。MacHack? はコンピュータチェスプログラムで、AI に批判的だったヒューバート・ドレイファスの鼻を明かしたこともある。ロボットブロック積みプログラム HACKER の原作者であるジェラルド・サスマンがいた。そして、ラボの数学の達人、ビル・ゴスパーがいた。彼は LIFE ゲームの哲学的な含意に触発されてハッキングにのめり込み、18 ヶ月目になろうとしていた。5

固く結び合ったこのグループのメンバーたちは、自分たちのことを「ハッカーズ」と呼んでいた。その内に、彼らはストールマンも「ハッカー」と呼ぶよ うになった。そうする過程で、彼らはストールマンに「ハッカー倫理」なる倫理的伝統を教え込んだ。ハッカーが魅了されていたのは、コンピュータができるこ との限界を探検することだった。挑戦に没頭していれば、36 時間でも、ぶっ通しで端末の前に座っていたかもしれない。最も重要なのは、（もし他に誰も使っていないなら）いつでもコンピュータにアクセスできること、 最高に有用な情報にアクセスできること、を要求したことだった。 ハッカーたちは、ソフトウェアによる世界の変革について開けっ広げに話し、ストールマンは、このハッカーの高貴な大義の遂行を妨げる障害に対する、ハッ カーの本能的な蔑みを身につけていった。こうし

た障害の最たるものが、下手なソフトウエア、大学の官僚主義、自己中心的なふるまいだった。

ストールマンはまた、障害に出会ったとき、ハッカーがいかに創造的な仕方でそれを回避したかについての伝承や物語を学んだ。これには、教授たちのオ フィスを開けて隠匿された端末を「解放する」ための様々な技が含まれていた。身勝手なハーヴァードの人々とは違い、MIT の教職員たちは AI ラボの端末を私物扱いするよりも良い扱い方を心得ていた。ある教職員が夜間に端末に施錠するという間違いを犯すと、ハッカーたちはすぐに再びその端末をア クセス可能にして、コミュニティが不当に扱われたことに抗議した。ハッカーの中には、これを鍵のピッキングでする者もいた（鍵ハッキング）。ある者は天井 のタイルを外し、壁を登った。ある者は、9 階を、コンピュータ・ケーブルのためのその仮フロアーも含めて、その洞窟探検下に置いた。「ぼくは実際に、ある教授のオフィスのドアをぶち破るのに使われ た、金属製の重い円柱のついた台車を見せられました、」とストールマンは言う。6

ハッカーの強硬さは、教授たちが自己中心的に障害を置き、ラボの仕事を妨げるのをやめさせるという有益な目的にかなっていた。ハッカーは人それぞれ の要望を無視しなかったが、それは他のすべての障害にならない方法でする必要があると主張した。たとえば、教授たちは、しばしば、オフィスには盗みから守 らないといけないものがあると言った。ハッカーは応えた。「オフィスを施錠したら誰も狙わないでしょう。でもそれは、全然、友好的じゃありません。ラボの 端末がその中にある限り、あなたは鍵で閉め出してはいけない。」

AI ラボの教員の数はハッカーよりもずっと勝っていたが、ハッカー倫理は浸透していた。ハッカーズは、コンピュータのパーツを設計し構築し、ユーザーが使うほ とんど全てのソフトウェアを書いている AI ラボの職員と学生だった。彼らは全てを常に動作するようにしていた。彼らの仕事は不可欠で、彼らは踏みつけにされることを拒絶した。ハッカーたちは、主要 なユーザーの依頼と並んでお気に入りのペットプロジェクトの作業をしたが、こうしたペットプロジェクトがめぐりめぐって、マシンやソフトウエア・プログラ ムをさらに改善することも稀ではなかった。中古車をチューンアップする 10 代の若者のように、ほとんどのハッカーたちは、マシンをいじり回すことが楽しくて仕方なかった。

このマシンいじりの衝動は何よりも、ラボの中心的な PDP-10 コンピュータにパワーを与えるオペレーティング・システムに向けられた。非互換時分割システム（Incompatible Time Sharing）を略して ITS と呼ばれるオペレーティング・システムは、まさしくその設計にハッカー倫理を組み込んでいた。ハッカーたちは、それを、プロジェクト MAC 起源のオペレーティングシステム、互換時分割システム（CTSS; Compatible Time Sharing System)、への抗議として構築し、それにならって命名した。当時、ハッカーたちは、CTSS の設計は余りに制限がきつく、プログラマが必要に応じてプログラムの内部構造を変えたり、改良したりする権能を制限していると感じていた。ハッカーたちに 代々伝えられている伝説によれば、ITS 構築の決定には政治的な意味も含まれていた。IBM 7094 のために

設計された CTSS とは異なり、ITS は特に PDP-6 のために作られた。 ハッカー自身にシステムを書かせることで、AI ラボの管理者たちはハッカーだけが PDP-6 を快適に使えることを保証した。学術研究の封建世界のなかで、この策略はうまくいった。PDP-6 は他の学部との共有物だったが、AI 研究者たちはまもなくそれを我が物とした。ITS と PDP-6 を土台として使い、ストールマンがやってくる少し前に、ラボはプロジェクト MAC からの独立を宣言することができた。7

1971 年までに、ITS は、PDP-6 を特別のスタンドアローンの利用のために残して、より新しい、しかし PDP-10 互換のものに移行していた。AI ラボの PDP-10 は、1971 年当時としては非常に大きな、1 メガバイト強に相当するメモリーを持っていた。1970 年代後期にそれは 2 倍になった。プロジェクト MAC は 2 台の別の PDP-10 をもたらしていた。それはどちらも 9 階にあり、すべて ITS が走っていた。ハードウェアが得意なハッカーは、これらの PDP-10 のための主なハードウェアの追加を設計し構築し、標準的な PDP-10 には欠けている機能、ページング方式の仮想メモリを実装した。8

見習いハッカーのストールマンは、すぐに ITS の虜になった。非ハッカーを寄せつけないようなところがあったが、ITS は、ほとんどの商用オペレーティングシステムが提供するのにあと数年かかるような機能、マルチタスク、デバッグ、フルスクリーンエディタのような機能を備 えていた。

「ITS は、あるプログラムが他のプログラムを調べるための、とても洗練された内部機構を持っていました、」とストールマンは、そのプログラムのことを思い出しな がら言う。「他のプログラムのあらゆる種類の状態を、とてもきれいで、うまく仕様化された仕方で調べることができました。」これはデバッギングに便利なだ けではなく、プログラムのスタート、ストップまたは他のプログラムのコントロールに便利だった。

もう一つ気に入った機能は、モニター中のプログラムのジョブを命令と命令の間で凍結できることだった。他のオペレーティングシステムなら、このよう な操作は、ユーザーからは見えず、良く意味が定義されない状態で、システムコールの途中でプログラムを止めるかもしれない。ITS では、それはプログラムのステップ実行をモニターするもう一つの方法を提供していた。

「『ジョブ停止』と言えば、それはいつもユーザモードで停止しました。 それは二つのユーザモードの命令の間で停止し、その瞬間のジョブに関する全てが、整合していました、」とストールマンは言う。「『ジョブ再開』と言えば、 それはちゃんと継続します。それだけじゃありません。（はっきりと見える）ジョブの状態を変更し、それを継続し、そして後にまた元に戻しても、矛盾は全く 生じないんです。隠れた状態は、どこにもありませんでした。」

1971 年秋に始まった、AI ラボでのハッキングはストールマンの週間予定の中で定例化していた。 月曜から木曜までは、ストールマンは起きている時間をハーヴァードの授業に捧げた。 しかし、金曜の午後が来るや、彼は電車に乗り、週末を過ごすために MIT へ向

かった。 ストールマンは到着時刻を、そこでの食事の儀式に合わせていた。 他の 5、6 人のハッカーによる夜毎の中華料理探訪に相乗りして、彼は、オンボロ車に飛び乗り、ハーヴァード橋を渡ってボストン近郊に向かった。その後の 2 時間は、ハッカー仲間と、ITS から中国語や象形文字のシステムの内部ロジックまで、あらゆることを議論した。夕食後、グループは MIT へ戻り、夜明けまでコードをハックするか、または再び午前 3 時に中華街に行った。

ストールマンは、毎日、朝までハッキングしていたかもしれないし、土曜の朝はカウチで眠っていたかもしれない。朝起きると、ストールマンはまた少し ハックし、中華料理の食事をとり、ハーヴァードに戻った。同様に、時々土曜日は、一日中ラボにいただろう。これらの中華料理の食事は美味であるばかりか、 ハーヴァードの大食堂では手に入らない生命維持の食事を提供するものでもあった。大食堂では平均して一食だけだったから、彼の食欲を満たせなかった。(朝 食は勘定に入らない。なぜなら、彼はたいてい朝食を好まなかったし、通常、その時間は眠っていたからだ。)

高校の仲間とはほとんどつき合わなかったオタクの外れ者が、突然、コンピュータと SF と中華料理という同じ偏向を有する人々と一緒に過ごすようになったことは、強烈な体験だった。「チャイナタウンからの帰りの車で何度も日の出を見たことを 思い出します、」とストールマンは、15 年後、スウェーデン王立工科大学でのスピーチで懐古している。「日の出を見るのは実に素敵なことです。それは一日の静穏なひとときだからです。これから ベッドに行けるというのは、一日の素晴らしい時間です。輝き始めた光を受け、鳥がさえずり始めたころに家路につくのはとてもいいものです。その夜なし終え た仕事について、穏やかな満足感と平穏を心から感じられるでしょう。」9

ハッカーたちとのつき合いが深まるにつれて、ストールマンはハッカーの世界観を受け入れるようになっていった。個人の自由という考えには以前から同 調していたが、ストールマンは自らの行動に共同体への責任感を染み込ませた。他の人が共同体の決まりを破ったら、ストールマンはすぐに遠慮なく指摘した。 最初に訪問した年から、ストールマンは施錠されたオフィスに侵入し、ラボのコミュニティ全体に属する、接収された端末を取り戻そうとした。本物のハッカー のやり方で、ストールマンは鍵ハックの技にも個人的に貢献しようとした。ふつうグリーンブラットに帰せられているが、もっとも芸術的なドア開けの小技のひ とつに、硬い針金を曲げてステッキ状にし、長い方の端にテープの輪を取り付けたものがあった。針金をドアの下から滑べり込ませ、ハッカーは長い方の端がド アノブへ触れるように針金をひねって回転させる。テープのひっかかりがあるので、ハッカーは外にある反対端から、できあがっているハンドルでドアノブを回 す事ができた。

ストールマンがその技を試してみると、実行が難しい事が分かった。テープをくっつけるのは簡単ではなかったし、ドアノブが回るように針金をひねるの も難しかった。ストールマンは、別の方法を考えた。廊下の天井に、ずらして外せるタイル部分がある。もしも飛び乗る机が一つあれば、これは常に使える。し かし、このやり方だと、ハッカーは大抵ガラ

ス繊維まみれになった。この欠点を直せる方法はないだろうか。ストールマンは別のアプローチを考えた。 ドアの下 から針金を入れるかわりに、天井のパネルを二枚外して、針金を持って手を伸ばしたらどうか。

ストールマンはそれを自分でやってみることにした。針金の代わりに、ストールマンは磁気テープを長い U 字型に垂らし、U 字の底のところに、粘着テープの輪を取り付けた。ドア枠を越えて手を伸ばし、テープを吊下げて、それがドアノブの下に巻きつくようにした。粘着力がしっかり働くところまでテープを持ち上げてから、彼はテープの端を引いて、ドアノブを捻った。案の定、ドアは開いた。こうして、ストールマンは鍵ハックの技に新手法を加えた。

「ときどきドアノブを回してからドアを蹴らないといけないことがありました、」とストールマンは言い、この新手法の欠点を回想する。「机の上の椅子に立って、うまくやるには幾らかバランスが必要だったんです。」

こうした活動は、政治信念を守るために発言し行動することに、ストールマンが意欲的になってきたことの反映だった。AI ラボの直接行動の精神に感化されて、ストールマンは、10 代の臆病で何もできない状態をうち破った。オフィスに侵入し、端末を解放するのは、抗議の行進に参加するのと同じではないが、大抵の抗議行動ではなしえな いような仕方で効果的だった。それは目の前の問題を解決した。

ハーヴァードの最後の年までに、ストールマンは、AI ラボの奇抜で不遜な教えを自分の学校でも適用しはじめていた。

「彼は、蛇の話をしたかしら、」と彼の母親はインタービュー中でこの点を尋ねた。 「彼と寮の仲間は蛇を学生選挙の候補に立てたんですって。けっこうな票を得たそうよ。」

ヘビはストールマンのいたカリア寮の候補の話で、キャンパス全体の学生委員候補の話ではなかった。ストールマンは、ヘビが相当な数の票を集めたの思 い出す。それは大部分、ヘビとその飼い主の姓が同じだったことのおかげだった。「みんな飼い主の方に投票するつもりで、ヘビに投票したんでしょう、」とス トールマンは言う。「選挙ポスターには、ヘビは『蛇の歩みで』オフィスに向かっていると書きました。ヘビは『逃走中の』候補だとも言いました。二、三週間 前に換気装置から壁に入り込んでしまって、行方不明になっていた[のでね/んだ;語調で。なぜなら、はなくてもいいかも]。」

ストールマンと寮仲間は寮監の息子を候補者に指名することもした。「彼の政治綱領は 7 歳になったら絶対辞める、でした」とストールマンは回顧する。とはいえ、これらのいたずらは、MIT キャンパスの偽候補のいたずらに比べると見劣りする。偽候補のいたずらでもっとも成功したのは、ウッドストックという名のネコで、キャンパス全体の選挙で 人間の候補を退けたのだ。「ウッドストックの得票は公表されず、無効票として処理されました、」とストールマンは回顧する。「でもこの選挙の無効票の多さ から見ると、ウッドストックは実際に勝っていたようです。二年後、ウッドストックは不審なことに車に轢かれてしまいました。運転手が MIT の行政関係者だったかどうかは不明です。」 ストールマンは、ウッド

ストックの立候補には関っていないが、「おそれいりました」と言っている。10

AI ラボでのストールマンの政治活動はより先鋭だった。1970 年代には、ハッカーたちは、ITS とそのハッカー寄りの設計を回避しようとする教授会や管理者たちからの、絶えざる挑戦にさらされていた。 ITS は誰でもコンソールの前に座ることができ、そして何でもすることができた。システムに 5 分でシャットダウンするよう命じることさえ可能だった。もし誰かが良くない理由でシャットダウンを命じたら、他の誰かがそれを取り消した。70 年代の中頃、一部の教授たちが (たいてい他の場所で彼らの姿勢を形成した人々だった)、データへのアクセスを制限するため、ファイルのセキュリティシステ ムを要求し始めた。他のオペレーティング・システムは、そういう機能を持っていた。そこで、それらの教授たちはセキュリティになじみ、それが何か危険なも のから守っているのだという気持ちだった。しかし、AI ラボは、ストールマンや他のハッカーたちの強い主張で、ノーセキュリティ圏に留まっていた。

ストールマンは、セキュリティの追加に反対する倫理的かつ実際的理由を述べた。倫理面では、ストールマンは、AI ラボのコミュニティの知的な開放性と信頼の伝統をアピールした。実際面では、ITS の内部構造は、全てのユーザーをコントロール下におくよりもハッキングと協力を育てるようにできていることを指摘した。この設計を逆転させるには大がかり なオーバーホールが必要だ。それをさらに難しいものにするため、彼は、どのユーザーが最後にファイルを変更したかを記録する機能のファイル記述子をどれも 最後の空領域まで使い切っていた。これは、ファイルのセキュリティ情報の余地を残していなかったが、誰も真剣にそれを除去する提案ができないので有益だっ た。

「ITS を書いたハッカーたちは、ファイル保護機能は、通常、自称システム管理者が他の全員に権力をふるうために使われると見なしていました、」とストールマンは 後に説明する。「彼らは、何人たりとも彼らに対してそうした権力を振るえるようにしたくなかったので、そういう機能を実装しませんでした。その結果、シス テムの中の何かが壊れても、いつでも自分で直せたんです。」(なぜなら、アクセスコントロールが立ちふさがることはないのだ。) 11

このような警戒心から、ハッカーたちは AI ラボのマシンをノーセキュリティのままにしていた。だが、近くの MIT コンピュータ科学研究所のあるグループでは、セキュリティ志向の教職員が勝利していた。DM グループは 1977 年に最初のパスワードベースのシステムを導入した。もう一度、ストールマンは、彼が倫理的堕落と見る事態を自ら正すことに決めた。パスワードシステムを制 御するソフトウエアコードへのアクセス権を入手して、システムが記録した暗号化されたパスワードを復号するプログラムを書いた。そして電子メールで、パス ワードを空白文字列を選ぶようにユーザーに求めるキャンペーンを始めた。あるユーザーが、例えば "starfish" (ヒトデ) と入力すると、こんなメッセージが帰ってきた。

「あなたは "starfish" と入力したようですね。パスワードをキャリッジリターンに変更

することをお勧めします。私もそれを使っています。こちらのほうがずっと入力が簡単で、パスワードとセキュリティのアイデアにも反対しています。」

実際に「キャリッジリターン」を入力したユーザー、つまり単に Enter キーやリターンキーを押し、ユニークなパスワードの代わりに空の文字列を入力したたユーザーは、かれらのアカウントを世界中からアクセスできるようにした ことになる。まさに全アカウントがこれまでそうであったように、そして間もなく、そうでなくなるものに。そこがポイントだった。アカウントにかけるピカピ カの新しい鍵でロックするのを断ることにより、彼らは鍵を持つというアイデアを嘲笑していたのだ。マシンに弱いセキュリティの実装をすることは、本物の侵 入者を防がないこと、それが問題ではないこと、なぜなら侵入者を懸念する必要がないこと、誰も侵入したがっているのではなく、ただ訪問したがっているのだ ということを彼らは知っていたのだ。

1984 年の本『ハッカーズ』のインタビューの中で、ストールマンは、LCS のスタッフの五人に一人がこの議論を受け入れ、パスワードを空にしたと、誇らしげに語っている。12

ストールマンの空文字列キャンペーンとセキュリティ一般への抵抗は、結局無駄に終る。1980 年代初頭には AI ラボのマシンですらパスワードベースのセキュリティシステムを誇示していた。 それでもなお、これはストールマンの個人的、政治的な成熟への大きな一里塚だった。 このことは、ストールマンのその後のキャリアの文脈から見て、生命を脅かす重大事について声を上げることも怖れる内気なティーンエイジャーから、 やがて辛言と甘言を業とするフルタイムの活動家への、わかりやすい転回点を示してくれている。

コンピュータセキュリティへの反対を表明するに当たって、ストールマンは、自分の初期の人生を形成した様々な力、知識への渇望、権威への嫌悪、一定 の人間を訳もなく爪弾きにする隠れた手続きや規則へのいらだち、を使った。彼はまた、彼の人生の後半を形成する倫理的概念、公共への責任、信頼、そして直 接行動のハッカー精神から力をひきだした。ソフトウエア・コンピュータ用語で言えば、この空文字列は、リチャード・ストールマンの政治的世界観バージョン 1.0 と言えるものだ。未熟な点はいくつかあるが、大部分は完全に熟している。

振り返ってストールマンは、自分のハッキング歴の最初期の出来事に、過大な意義を与えることを躊躇する。「初期段階では、多くの人が私と同じ気持ち を共有していました、」と彼は言う。「パスワードを空文字列にした人が沢山いたことが、多くの人々がそれを正しいことだと同意したしるしです。私は、それ について活動したいと思っただけでした。」

それでも、ストールマンは、活動家精神を覚醒させたのは、AI ラボだとする。十代のストールマンは政治的事件を見ても、一個人が重要なことをしたり言ったりできるとは夢にも思わなかった。青年になったストールマン は、彼が最も信頼できると感じる事柄、ソフトウエア設計、公共的責任、個人の自由といった事柄について声を上げるようになった。「ぼくは、お互いの自由を尊重する暮らし方を持つコミュニティに参加しました、」と彼は言う。「[ぼくは/;省略]すぐにこれが良いことだと判りました。 これがモラルの問題である

と結論するにはもう少し時間がかかりました。」

AI ラボでのハッキングだけがストールマンの自信を強めるのに役立った活動ではない。ハーヴァードでの学部 3 年の中頃に、ストールマンは、カリアーハウスで始まったばかりの娯楽の国際的な民族舞踊グループに参加した。彼は自分にダンスはできないと思って、やろう としないでいたが、ある友人が「やってみないと、できるかどうか分からないさ」と指摘した。自分でも驚いたことに、彼はうまく踊れて、ダンスを楽しむこと ができた。試しに始めたことだが、すぐにハッキングと研究に並ぶほどの情熱を傾けるものになった。それは、しばしば女性と出会うきっかけにもなった。カ レッジ生活の間はデートまで行かなかったが。踊っているストールマンは、フットボールで挫折した、臆病で、不器用な 10 歳の時のようには感じていなかった。彼は自信を持ち、敏捷で、生き生きしていると感じていた。80 年代の初め、ストールマンはさらに進んで、MIT 民族舞踊団（the MIT Folk Dance Performing Group）に加わった。観客のために踊り、バルカン農民の伝統的衣装の偽物で着飾り、観客を前にすることが楽しいことだと分かって、彼は後に演説すると きに助けとなるステージでの適性を発見した。

ダンスとハッキングはストールマンの社会的地位の向上には役に立たなかったが、彼のハーヴァード以前の生活を曇らせていた追放感を克服するのには役 立った。1977 年に初めて SF 大会に参加したとき、彼はボタン製作者のナンシーに出会った。何でも希望する言葉で筆記体文字のボタンを作ってくれる。ストールマンは興奮して「神を弾劾 せよ」という言葉で飾られたボタンを注文した。

ストールマンにとって、「神を弾劾せよ」というメッセージはいろいろなレベルで影響した。 幼い時からの無神論者であるストールマンは、まずそれが、進行中の宗教論議に「第二戦線」を設定する試みに見えた。「あのころは、誰もが神が実在するかど うかを論じていました、」とストールマンは回想する。「『神を弾劾せよ』は神の問題に全く別の視点からアプローチしていました。神様が世界を創造するほど の力を持ちながら、その問題を正さないなら、 なんで我々は崇拝したいと思うのでしょうか。裁判にかけたほうが良いんじゃありませんか。」

それと同時に、「神を弾劾せよ」はアメリカとアメリカの政治システムに対する風刺だった。1970 年代のウォーターゲート事件はストールマンに深く影響した。子供のころから、彼は権威に対する不信のなかで育ってきた。今、大人として、彼の不信は、AI ラボのハッカーコミュニティの文化によって確固たるものになった。ハッカーたちにとって、ウォーターゲートは、特権を持たない者の生活を困難にしている 日々の権力闘争を、シェークスピア風に上演してみせたものにすぎなかった。 それは人々がセキュリティと便利のために自由と寛容を売り渡すときに起こることの特大サイズの寓話だった。

育った自信に支えられて、ストールマンはそのボタンを誇らしげに身につけた。 好奇心から彼にそのことについて尋ねた人々は、前もって準備していた口上を聞かされた。「我が名はエホバ、」とストールマンは言った。

「私には不正と苦しみを終わらせる秘密の計画がある。 しかし天国のセキュリティ要件により、 その計画の活動を君に教えることはできない。 私にはその状況が見えるが、君には見えない。 君の知るように私は良き者である、なぜなら私がそう言うからだ。 君はただ私を信じ、質問はせず、従わなければならない、 もし私に従わないと、きみは悪魔ということだ。 だから、君を私の敵のリストに載せ、地獄に投げ込んでやろう。 そこで地獄の税務所から永劫の監査を毎年受けるのだ。」

この口上を、ウォーターゲート事件公聴会のパロディだと取った人は、 このメッセージの半分しか聞いていない。 ストールマンにとって、残りの半分は、ハッカー仲間にだけは聞こえるような何かだった。 絶対的権力は絶対に腐敗する、とアクトン卿が警告してから百年後、アメリカ人はアクトン卿の自明の理の前半を忘れたように見える。 権力というのは本来腐敗していくものなのだ。 数え切れない小さな腐敗の例をあげつらうより、ストールマンは、そもそも権力を信用するシステム全体への憤慨を表明することに満足感を覚えた。

「なぜこの小さなハエでやめるのかと、思いましたね、」と、ボタンとそのメッセージを思い起こしながら、ストールマンは言う。「ニクソンを追い回す のなら、なぜミスタービッグを追い回さないんでしょう。ぼくの見方では、権力を持ちながら、それを濫用する存在は、いかなるものでも、その権力を剥奪され るべきなんです。」

後注

1 Michael Gross, "Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur?-certified Genius" (1999); マイケル・グロス『リチャード・ストールマン：高校の落ちこぼれ、 自由なソフトウェアのシンボル、マッカーサーお墨つきの天才』(1999年)を参照。

2 カルミネ・デサピオは、ニューヨーク市の政治的支配集団タマニー・ホール の初代のイタリア系アメリカ人のボス、ということで悪名高い。 デサピオについてもっと知りたければ、 John Davenport, "Skinning the Tiger: Carmine DeSapio? and the End of the Tammany Era," New York Affairs (1975): 3:1 ジョン・ダベンポートの 『虎の皮を剥ぐ：カルミネ・デサピオとタマニー時代の終焉』ニューヨーク・アフェアズ(1975 年): 3:1 を参照。

3 コロンビア・サイエンス・オナーズ・プログラムのもう一人の出身者であるチェスは、こうした抗議行動を「背景の雑音」と形容する。 「政治に関心が無いわけではありませんでしたが、」と彼は言う、 「SHP は大事だったんです。それをすっ飛ばしてデモに行こうとは思いませんでした。」

4 しかし、ストールマンはこれを疑う。「数学と物理からプログラミングに移った理由の一つは、ぼくは前の二つで何か新しいものを発見する方法を学べなかったからです。他人が何をしたのか学んだだけでした。プログラミングでは、毎日何か有益なことができました。」

5 Steven Levy, Hackers (Penguin USA [paperback], 1984): 144; スティーヴン・レビー

著、「ハッカーズ」（工学社）182 ページ参照。レビーは、イギリスの数学者、ジョン・コンウェイが最初につくった数学ベースのソフトウェア・ゲーム LIFE に魅了されるゴスパーの描写に 9 ページを捧げている。私はこの本を追加情報として心からお勧めしたい。この本は必読書というべきかもしれない。

6 MIT の教授会のメンバーで、ストールマンより早くから AI ラボで働いていたハッカーのジェラルド・サスマンは、この話に異論がある。サスマンによれば、端末を救い出すためにハッカーがドアを破ったということは決してない。

7 前掲書

8 いまだに多くのハッカーが、ハッカー精神の縮図と見ている ITS の創世紀を駆け足で要約してしまって申し訳ない。このプログラムの政治的な意義についてもっと知りたい向きは、 Simson Garfinkel, Architects of the Information Society: Thirty-Five Years of the Laboratory for Computer Science at MIT (MIT Press, 1999) シムソン・ガーフィンケル『情報社会の建築家たち：MIT コンピュータ科学研究所の 35 年』（MIT Press、1999 年）を参照されたい。

9 リチャード・ストールマン『RMS KTH 講義(スウェーデン)』(1986 年 10 月 30 日）を参照。 http://www.gnu.org/philosophy/stallman-kth.html

10 本書が編集の最終段階に入って間もなく、 ストールマンは、電子メールで、ハーヴァードのキャンパスでも、 政治的なひらめきを得たと言ってきた。「ハーヴァードの初年度に、中国史の授業で、秦王朝に対する最初の反乱について読みました、」 と彼は言う。（全ての本を焼き、素焼きの戦士とともに葬られた残酷な建国者の国の話だ） 「この物語は歴史としては信用できませんでしたが、感動しました。」

11 リチャード・ストールマン（1986）

12 Steven Levy, Hackers (Penguin USA [paperback], 1984): 417; スティーヴン・レビー著、「ハッカーズ」（工学社）592 ページを参照。

第五章　自由の小さい水たまり

［RMS: この章は、私の考えと感情の事実を含む、事実に関する主張を訂正し、出来事の記述の中にある要らざる敵意を除いた。注記したもの以外、ウィリアム自身の印象の主張は保存されている。］

リチャード・ストールマンと一分でも一緒に過ごした人に聞いてみよう。同じ思い出が集まるだろう。長髪も奇矯な振る舞いも無視してかまわない。いきなり気が付くのは凝視だ。ストールマンの緑の瞳をのぞきこんだ人は、本当に信念の人の前にいることを実感する。

ストールマンの視線を強いの一言で片付けてしまうのは言葉が足りない。ストールマンの瞳は見つめるだけではない。貫くのだ。霊長類特有の礼儀でちょっと目をそらしているときでも、ストールマンの瞳は固定されていて 2 条の光子線のように側頭部をじりじりと焼く。

おそらくこのせいだろう。多くの記者がストールマンを描写するときに宗教的角度から書く傾向がある。「自由ソフトウェアの聖者」と題した 1998 年の Salon.com の記事でアンドリュー・レオナルドは、ストールマンの緑の瞳を「旧約聖書の預言者の力を放っている」と形容した。1 1999 年のワイアード誌の記事は、ストールマンのあごひげを「ラスプーチンのような」と形容したし、2　ロンドン・ガーディアン誌の横顔紹介は、ストールマンの微笑は「イエス様を見ている使徒」の微笑と書いている。3

そういう類推は役立つものなのだが結局うまくいかない。なぜなら、それはストールマンの人格の傷つきやすい面を見ていないからだ。ストールマンの視 線を長時間観察していると微妙な変化に気付き始める。最初は威嚇や催眠効果を狙ったように見えたものが、2 度 3 度と見ていると、接触を図り接触を保とうとして挫折した試みだということが分かってくる。もしもストールマン自身が時々疑ったように、彼の性格が自閉症か アスペルガー症候群の産物だとすれば、彼の視線こそ診断を裏付けるものだろう。目の光が最も強い瞬間にも曇ったり遠くなったりしがちである。傷ついて死を 覚悟しようとしている動物の目のようだ。

私自身が伝説的なストールマンの凝視に遭遇したのは、1999 年 3 月のカリフォルニア州サンノゼで開催されたリナックスワールド・コンヴェンション・アンド・エクスポに遡る。Linux ソフトウェア・コミュニティの「お披露目パーティー」と宣伝されたこのコンヴェンションは、ストールマンを科学技術メディアに再び紹介するイベントとして も際立っていた。クレジットの正しい割合を強く要求することに決めたストールマンは、GNU プロジェクトの歴史と明確な政治的目標を観客と記者に教えるた めにこのイベントを使った。イベントをカバーするために送り込まれた記者として、グラフィカル・ユーザ・インタフェースの自由ソフトウェア、 GNOME（グノーム）1.0 のリリースを告知する記者会見中に私はストールマンの授業を受けた。ストールマン本人に投げた、ちょうど最初の質問で、それ

と知らずに議論の引き金をひいてしまったのだ。「GNOME の成熟性は、Linux オペレーティング・システムの商業的な人気に影響すると思われますか？」

「このオペレーティング・システムを Linux と呼ぶのは、どうぞおやめ下さい。」と、ストールマンは応じ、目の焦点はすぐに私の目に合っている。「Linux のカーネルはオペレーティング・システムの小部分にすぎません。みなさんが Linux と呼んでいるこのオペレーティング・システムをつくっているプログラムの多くは、リーナス・トーバルズが開発したものではありません。GNU プロジェクトの有志の手で、ユーザーが今ある自由なオペレーティングシステムを持てるように、個人的な時間を費やして創造してきたものなのです。これらの プログラマの貢献を認めないことは、失礼であり、歴史を誤って伝えることです。これがオペレーティング・システムに言及するときにはその固有名詞、 GNU/Linux（グヌー・リナックス）と呼んでくださいと皆さんに求める理由です。」

この言葉を取材手帳に書きながら不気味な沈黙が部屋を支配するのに気づいた。ついに私が顔を上げると、ストールマンが瞬きもせずに私の返事を待って いたのがわかった。こわごわと、二番目の記者がリナックスのかわりにグヌー・リナックスを使うように気をつけて質問した。GNOME プロジェクトのリー ダー、ミゲル・デ・イカサ（Miguel de Icaza）がその質問をさばいた。イカサが半分くらい答えるまで駄目だったが、ついにストールマンの目は私から離れた。そのとき、軽い戦慄が走った。ス トールマンが気付いた用語の誤りについて他の記者にレクチャーを始めたとき、いけないことを言ってしまったが解放されたと感じた。少なくとも彼は見ていな いと私は自分に言ったのだった。

ストールマンにとって、こういう顔を付きあわせる時間は目的に役立っている。最初のリナックスワールド・ ショーが終わるまでに、大半の記者は Linux という用語をストールマンの前で使うよりも良いことを知ったし、ワイアードはストールマンをスターリン以前の革命家になぞらえて、GNU プロジェクトの政治的な目標を重視しすぎないように熱望するハッカーや起業家の手によって、歴史の本から消された革命家として描く記事を掲載している。4　ほかに追随する記事でこのオペレーティング・システムを GNU/Linux と表記する記者は少なかったものの、ほとんどの記事は、ストールマンが自由ソフトウェアのオペレーティング・システムの構築を 15 年前に開始したことを直ちにクレジットした。

それから 17 か月、私はストールマンに会うことはなかった。その間、ストールマンは 1999 年のリナックスワールド・ ショーに行くために再びシリコンバレーを訪れた。ストールマンは、講演者としては招待されていなかったが、このイベントで最高のセリフを口にした。コミュ ニティへの貢献により、このショーで Linux の創始者の名前をつけたリーナス・トーバルズ賞を自由ソフトウェア財団を代表して受けとったときの皮肉だ。 「自由ソフトウェア財団にリーナス・トーバルズ賞を贈るのは、反乱軍にハン・ソロ賞を贈るのにちょっと似ているね。」

しかし今回のコメントはメディアに大きな反響を起こせなかった。その週の半ばに、GNU/Linux の著名な販売者であるレッドハット（Red Hat, Inc.）の株式公開があった。こ

のニュースは、たとえば、私のような記者の多くが薄々感じていたことを裏付けたものにすぎなかった。つま り、"Linux"は、以前の「e コマース」や「ドットコム」とそっくり同じウォール街の流行語になったのだ。2000 年問題を前にした入れ換え需要が近 づいて、株式市場が垂直的漸近線に近づく双曲線を描いて上昇するとともに、自由ソフトウェアやオープンソースを政治的現象として語る試みはすべて道端へと 追いやられた。

たぶんそれが、前 2 回に引き続き開催された 2000 年 8 月の第 3 回リナックスワールド・ ショーに、目立つような形でストールマンが欠席した理由だ。

私の二度目の、ストールマンと彼のトレードマークの凝視との遭遇は、その第 3 回リナックスワールド・ ショーのすぐ後だった。ストールマンがシリコンバレーに行くことを聞きつけた私は、カリフォルニア州パロアルトでランチ・インタビューを設定した。先日の ショーをすっぽかしたことだけではなく、全般的な背景から、その場所は風刺的な感じがした。ワシントン州レドモンドを別とすれば、ここほど占有的ソフト ウェアの経済的価値の直接的証拠を提供している都市はない。貪欲と自己中心主義を偏愛する我々の文化と戦うことに人生の大半を費やしてきたストールマン は、ガレージぐらいの大きさの平屋の不動産を 50 万ドルの価格帯で取り引きするような街でどうしているんだろう、見に行かないとと思いながら、私はオークランドから車を走らせていた。

ストールマンの教えてくれた道順に従い、非営利の「バーチャル・アーティストの共同体」、アート・ドット・ネット（Art.net）の本部に着い た。街の北部の生け垣に覆われた家にあるアート・ドット・ネットの本部は清々しく荒廃していた。急に、シリコンバレーの中心にストールマンが潜んでいると いうアイデアも、結局、たいして不思議なこととは感じられなくなった。

ストールマンは暗い部屋に座り、ラップトップコンピューターに入力しているところだった。私が部屋に入っていくと、すぐ顔を上げて 200 ワットの眼光を投げてきた。人を落ち着かせる「ハロー」という挨拶をくれたので、私も挨拶を返した。しかし、それが声になる前に彼の目はラップトップの画 面に戻っていた。

「ちょうどハッキングの精神の記事を書き終えるところです」と言って、指はまだキーを叩いていた。「見て下さい。」

私は見た。部屋は薄明かりで、文章は黒の背景に緑がかった白い文字で表示されていた。多くのデスクトップ・ワープロの画面のちょうど逆の配色に慣れ るまで少し時間がかかった。目が慣れると、私は、ストールマンが最近食事をした韓国料理店の記事を読んでいた。食事の前に、ストールマンは興味深い発見を する。給仕をした人は、ふつう箸は 2 本なのに 6 本もストールマンの席に置いていった。レストランのたいていの客は余分な箸を無視するところだが、ストールマンはこれを課題と見なした。6 本すべてを一度に使えという課題だ。多くのソフトウェア・ハックと同じく、うまい解決策は巧妙であると同時に馬鹿馬鹿しい。そこで、ストールマンはそれを イラストレーションとして使うことに決めた。

記事を読んでいるあいだ、ストールマンがじっと私を見ているのを感じた。通読していて、

自慢気だが子供のように顔が微笑んでいるのに気付いた。私がよいしょして記事を持ち上げると、少しだけ彼は眉を上げてくれた。

「出かける用意はすぐにできます」と彼は言う。

ストールマンはまたラップトップを叩きはじめた。箱型で灰色のラップトップは、先日のリナックスワールド・ショーでプログラマに人気だった洒落た現 代的なラップトップとは似ていない。キーボードの上に、ストールマンの老いつつある手を証明する小さくて軽いキーボードが載っていた。1990 年代の中頃、手の痛みが堪えきれないほどになってタイピストを雇わなければならなくなった。現在、ストールマンは一般的なキーボードに比べて押す力が少な くてすむキーボードに頼っている。

ストールマンは仕事をするときに外界からの刺激をすべて遮断する傾向がある。彼の目が画面に釘付けになり指が躍っているのを見ると、古い友人どうしが会話に夢中になっているかのようだ。

何度かキーストロークの大きな音を立て、ゆっくりとラップトップを閉めてセッションが終わった。

「昼食の準備はいいですか。」とストールマン。

私たちは私の車まで歩いた。足首が痛いと訴えてストールマンはゆっくり足をひきずっていた。左足の腱の怪我のせいだという。3 年前の傷だが、悪化して大好きなフォークダンスを踊るのを完全にやめざるを得なくなってしまった。「生まれつきフォークダンスが好きなんですね。踊ること ができないのはぼくの悲劇ですよ。」とストールマンは嘆く。 ストールマンの体が悲劇を物語っている。運動不足で頬はふくらみ、1 年前にはそれほど目立っていなかった腹も出ている。体重の増加が劇的だったことはすぐにわかる。ストールマンが歩くとき、慣れない重みと折り合いをつけよ うとする妊婦のように背中を弓なりにそらせるからだ。

歩みは、ストールマンが文字通り立ち止まってバラの香りをかごうとしたために、さらに遅くなった。とくに美しい花を指さし、大きな鼻で一番奥の花びらに触れて深く息を吸い込み、満足げな吐息で後ろに下がる。

「うーん、リノフィートフィーリア（rhinophytophilia）」と言って背中をさすった。5

レストランは車で 3 分もかからなかった。自由ソフトウェア財団の前事務局長、ティム・ネイ（Tim Ney）に勧められて、私は店の選択をストールマンに頼んでいた。修道僧のような生活に焦点を合わせる記者もいるが、ストールマンは食事のときに実は快楽 主義に染まっている。自由ソフトウェアの大義を広めるために世界を飛び回る使節の付加給付の一つは、世界中の美食を体験できることだ。「世界のほとんどの 大都市を訪問して、その街で一番のレストランを知るチャンスがあった」とネイは言う。「リチャードは、メニューに載っているものがどんなものか知ってい て、テーブルいっぱいになるくらい注文するのが自慢なんだ。」（つまり、みんながその気なら。）

今日の食事にストールマンが選んでいたのは、パロアルトの目抜き通りから 2 ブロック

入ったところにある広東風の点心レストランだった。この選択は、部分的には、講演で広東省に滞在した最近の中国訪問にインスパイアされたものだ。 好みから言うと、ストールマンは湖南や四川の香辛料をきかせた料理が苦手である。「スパイシーな料理の大ファンではありません」とストールマンは認める。

私たちは 11 時を数分過ぎて到着したが、すでにもう 20 分待ちの状態だった。ハッカーは時間の無駄使いを嫌悪することを知っていたので、一瞬、怒るんじゃないかと固唾を呑んだ。予想とは逆に、ストールマンはこの情報をあっさり受け止めた。

「ほかに食事の仲間が見つからなくて残念でしたね」と彼は私に言う。「いつだって食事は大勢で食べたほうがもっと楽しい。」

待っている間に、ストールマンはダンスのステップを少し練習する。彼の動きは、ためらいがちだったが、見事なものだった。私たちは最新の出来事を話 題にした。ストールマンが言うには、リナックスワールドに出なくて残念だったのは、GNOME 財団の発足を発表する記者会見に出られなかったことだけだと言う。サン・マイクロシステムズと IBM の後援により発足したこの財団は、自由ソフトウェアと自由市場経済は互いに排斥するものではないと言ってきたストールマンの主張が正しいことを色々な意味 で証拠づけるものだ。それにもかかわらず、ストールマンは発せられたメッセージに満足していなかった。

「あの発表の仕方はどうなんでしょう、会社が Linux のことを話すときに GNU プロジェクトに全然言及しませんでした」とストールマンは言う。

そんな失望感は、海外からの、とくにアジアからの暖かい反応とのコントラストを強く感じさせただけだったとストールマンは言う。ストールマンの 2000 年の旅程表を見れば一目瞭然だが、自由ソフトウェアのメッセージは人気を広げ続けていることがわかる。このところのインド、中国、ブラジルへの旅行の合間 にアメリカの土を踏んだのは、115 日のうち 12 日しかない。ストールマンは旅行のおかげで自由ソフトウェアの概念が異なる言語の文化にどう翻訳されるか 理解する機会を得た。

「インドでは多くの人が自由ソフトウェアに関心を持っていますが、大金をかけなくても彼ら自身のコンピュータのインフラを作りだせるやり方だと見て いるからです」とストールマンは言う。「中国ではこの概念が普及するのにずっと時間がかかっています。自由ソフトウェアを言論の自由になぞらえて考えるの は、そもそも言論の自由がないところでは難しいんでしょう。それでも、最近訪問したときの自由ソフトウェアに対する関心の水準は高かったですね。」

会話はナップスターに移る。ナップスターは、カリフォルニア州サン・マテオのソフトウェア企業で、この数ヶ月で一躍メディアの注目を集めた会社だ。 音楽ファンがほかの音楽ファンの音楽ファイルを閲覧してコピーできることで話題のソフトウェア・ツールを市場に出している。拡大するインターネットの力の おかげで、このいわゆる「ピア・トゥ・ピア（peer-to-peer）」のプログラムは、オンラインのジュークボックスとして事実上の標準に成長した。 その結果、一般の音楽ファンが使用料を支払わずに MP3 の音楽ファイルを聴

くことができるようになり、レコード会社は大いに悔しがることになった。

ナップスターのシステムは、占有的ソフトウェアを基盤にしているが、ストールマンの古い持論の論点からインスピレーションを得ている。すなわち、作 品がいったんデジタルの領域に入れば、つまり、いったん、コピーが音の複製や原子の複製以下のものになり、情報の複製以上のものとなれば、作品を共有した いという人間の自然な欲求を制限することは、より困難になっていくだろう。ナップスターの経営陣は、制限を加えるよりもこの欲求を活用することに決めた。 音楽リスナーに音楽ファイルを交換する中心的な場を与えて、その結果として生じるユーザートラフィックがほかの商機を生んでくれる可能性に賭けたのである。

ナップスターモデルの突然の成功が、伝統的なレコード会社に恐怖を植えつけたのも無理はない。私がパロアルトでストールマンと会う数日前は、全米レ コード協会の請求したファイル交換サービスに対する差し止め命令を連邦地方裁判所のマリリン・パテル（Marilyn Patel）判事がちょうど認めたところだった。この差し止め命令は連邦第九控訴裁判所によって執行を停止されたが、2001 年の初頭までには、控訴裁判所もサンマテオに本社を置くこの会社が著作権を侵害したと認めることになろう。6 この決定については、同協会のスポークスマン、ヒラリー・ローゼン（Hillary Rosen）が後に、「創造的なコンテントのコミュニティと合法的オンライン市場の明白な勝利だ」と宣言することになる。7

ストールマンのようなハッカーにとっても、ナップスターのビジネスモデルは別の意味で恐るべきものだ。この会社は、ファイルの共有と情報の共有とい うハッカーの古びた原則を流用するのに熱心だが、同時に、占有的ソフトウェアに基づくサービスを売っているので、悩ましい曖昧なメッセージになっている。 ストールマンは注意深く明確に表現されたメッセージをメディアに流すのに十分苦労してきた人間だから、この会社について話せと言われても発言が控えめにな るのは無理もない。しかしそれでも、ストールマンは、ナップスター現象の社会的側面からいくつか学んだことがあると認める。

「ナップスター以前には、人々が娯楽作品をプライベートに再配布することで[充分]と考えていました。」とストールマンは言う。「でも、ナップス ターが役に立つと思った人たちに教えられたのは、知り合いの間だけでなくて、ひろく公衆に向けて再配布する権利は本質にかかわるものだということです。し たがって、その権利を取り上げてはいけないんです。」

ストールマンがこれを言うやいなや、レストランのドアが開き、案内係に招き入れられた。そしてあっという間に、レストランの大きな鏡張りの壁ぎわの隅の席に座っていた。

レストランのメニューは注文用紙も兼ねていて、ストールマンは案内係がテーブルに水を持ってくる前から、さっさとマスを埋めていく。「豆腐の皮で包んだ海老のフライ」とストールマンが読み上げる。「豆腐の皮。おもしろい質感のものを出すんだね。これは食べてみるべきだ。」

この言葉から、中華料理とストールマンの最近の中国旅行に関する即興的な議論が始まった。「中国の食べ物は、ほんとに絶妙です」と言うストールマン の声は、この日初めて感

情の色を帯びた。「アメリカじゃ見たこともないものが本当にたくさんあるんです。地元で取れたキノコに地元の野菜で作った地元の食 べ物。毎回、食事が素晴らしくて、その記録を全部取る目的だけで日記をつけ始めたんですよ。」

会話は切れ目なく続いて韓国料理の話になった。同じ 2000 年 6 月のアジアへの旅行のときに、ストールマンは韓国を訪れた。彼が到着してから、地元のメディアにちょっとした旋風が巻き起こった。ちょうど同じ週に、マイ クロソフトの創業者のビル・ゲイツ会長が出席する韓国ソフトウェア・カンファレンスがあったからだ。その旅行中、ソウルの最有力紙の第一面のゲイツの写真 の上にストールマンの写真があったことの次に良かったのは、食事だと言う。「冷麺を食べたんです。冷麺というのは冷やして食べる麺です。」とストールマン は言う。「とてもおもしろい食感の麺です。どこでも同じ種類の麺を使うわけじゃないから、これこそ、これまでで最高の絶妙な冷麺だと断言できるんです。」

「絶妙（exquisite）」という用語は、ストールマンが絶賛するときの言葉だ。ストールマンが冷麺について熱狂的に語るのをしばらく聞いた後だから良くわかる。彼の視線のレーザー光線が私の右肩の表面を焦がすのを感じた。

「ちょうど後ろに、最高に絶妙な女性が座っていますよ。」とストールマンが言う。

振り返ると、女性の背中がちらっと見えた。若い女性で 20 代半ばぐらい、スパンコールの白いドレスを着ていた。彼女と連れの男性は支払いを終える段階だった。そちらを見なくても、2 人がレストランを出ようと席から立ち上がったのがわかった。ストールマンの視線の強さが急に弱まったからだ。

「あーあ、」とストールマン。「行ってしまいました。もう二度と出会えないんでしょうね。」

短いため息をついて、ストールマンは気を取り直す。その瞬間が、ストールマンの女性に関する評判を議論する機会を与えてくれた。評判は時々矛盾して いる。多くのハッカーが、ストールマンは女性の手の甲にキスをする挨拶を偏愛していると伝えている。8 その一方で、2000 年 5 月 26 日の Salon.com のある記事はストールマンをハッカー版の女たらしとして描いている。記者のアンナリー・ニューヴィッツ（Annalee Newitz）は自由ソフトウェアと自由恋愛の関連を論じ、ストールマンを伝統的な家族の価値を否定する人物として紹介して、「愛は信じていますよ。一夫 一婦制は信じていません。」という彼の言葉を伝えている。9

私がこの話を持ち出すと、ストールマンはメニューを少し下に下げた。「そうですね、たいていの男性はセックスをしたがってるように見えるし、どちら かというと、女性をばかにした態度をとってるように見えます。」と彼は言う。「つきあってる相手に対してもですよ。全く理解できません。」

私は、1999 年の『オープンソース』という本の一節に言及した。その中でストールマンは、不運な GNU カーネルに、当時の恋人にちなんだ命名をしたかったと告白している。彼女の名前はアリックス（Alix）で、新しいカーネルの名前の末尾に x をつける（例えば、Linux ）Unix 開発者の慣習にぴったりだった。彼女は Unix システムの管理者だったが、

友達に「誰かが私の名前をカーネルに付けるべきよ」と言い出したことがあった。そこで、ストールマンは彼女へのサプライズに、 GNU カーネルの名前を Alix にすることに決めた。カーネルの主要開発者がカーネル名を Hurd に改名してしまったが、カーネルの一部分には Alix の名前が含まれていた。アリックスの友達の一人が、ソースのスナップショットの中にあるこの部分に気付いて彼女に教え、彼女は感激した。後に再設計された Hurd では、その部分は削られている。10

この日はじめてストールマンがほほえんだ。私は手の甲にキスをすることを持ち出した。「ええ、そうしますよ。」とストールマンは言う。多くの女性が喜んでくれる愛情の表現方法の一つだと思うんです。好意を示し、それを認めてもらえる良い機会です。」

愛情はストールマンの人生にくっきりと走っている糸だ。そしてその質問が出ると痛ましいくらい率直になる。「自分の心の中を別にすれば、実際、ぼく の人生の中にたくさんの愛情はありませんでした。」と彼は言い、しかし、議論はすぐにぎこちないものになっていく。一語だけの返事をいくつか返して、つい にストールマンはメニューを持ち上げて質問をさえぎる。

「シューマイはいかがですか。」

料理が出てくると、会話は出てきた料理をスラロームする。よく言われるハッカーの中華料理好きについて議論し、ストールマンが AI ラボのプログラマだったときに週に 1 回のディナーはボストンの中華街に駆け込んでいた話や中国語に内在する論理とそれに関係する文字体系の議論をする。私のつっこみは、ストールマン側の精通 した議論でかわされてしまう。

「この前、中国に行ったときに上海語でしゃべる人達の話を聞いたんです。」とストールマンは言う。「面白かった。音が、全然［北京語と］違うんです ね。語源が共通の上海語と北京語の言葉をいくつか話してもらいました。類似点が分かるものもありますが、不思議に感じた一つの疑問は、どの音が同じなのか ということです。同じではないんです。興味深いことでした。というのも、音は、失われたり置き換わったりする追加的シラブルから進化するという理論があり ます。それらの影響は音の中に残るといいます。もしそれが本当なら、そして有史時代にそれが起きたという主張を見たことがあるんですが、方言は、これらの 最後のシラブルが失われる前に分岐したにちがいありません。」

最初の料理、大根モチがやってきた。ストールマンと私は二人で少し時間をかけて大きな長方形の大根モチを切り分けた。それはゆでキャベツのような匂いがしたが、ジャガイモのラートカをベーコンと焼いたような味がした。

私は、十代の頃の経験がストールマンを人気のない立場をとらせるのか、という見捨てられた話題をもう一度持ち出してみることにした。有名なのは、 1994 年以来コンピュータ・ユーザーとメディアにリナックスではなくグヌー・リナックスという語を使わせようとしている苦闘である。

「[のけ者であることは、人気のある見方に従うのを回避することに]役に立っていたと思

います。」と、餃子を噛みながら、ストールマンは言う。「ぼ くは他人に対する仲間の圧力の作用というものが全然わかりませんでした。それがあきれるほど拒絶されていた理由なんだと思います。一時的流行を追っても、 何もそこに得るものはありません。それは何の違いも生みださないでしょう。拒絶されたままでしたから、流行を追ってみようともしませんでした。」

ストールマンは、彼の逆張り傾向の手掛かりになる例として、音楽の好みを指摘した。一人のティーンエイジャーとして、高校の同級生がほとんどモータ ウンやアシッドロックを聴いているときに、ストールマンはクラッシック音楽を好んだ。その思い出は、中学時代のストールマンの珍しいユーモラスなエピソー ドに導く。ビートルズが 1964 年のエド・サリバン・ショーに出現して以来、ストールマンのクラスメイトは、ビートルズの最新アルバムや最新シングルを買いに走っていた。その時、流行の 四人をボイコットすることに決めたとストールマンは言う。

「ビートルズ以前のポピュラー音楽が好きでした」とストールマンは言う。「でも、ビートルズは好きじゃありませんでした。とくに嫌いだったのは、彼らに反応する人たちの野蛮な騒ぎぶりです。まるで、ビートルズにお世辞を言うためにビートルズ集会をやるみたいでした。」

ビートルズのボイコットはうまくいかなかったので、ストールマンは仲間たちの群れ意識を指摘するほかの方法を探した。ストールマンは、しばし、リバプール出身のグループを風刺するバンドを自分でつくろうと思ったと言う。

「それを東京ローズとジャパニーズ・ビートルズと呼びたかったんです。」

彼はいま、世界各地のフォーク・ミュージックを愛好しているので、ボブ・ディランや 1960 年代の他のフォーク・ミュージシャンも同じように好き だったのか聞いてみた。ストールマンは首を振る。「ピーター・ポール・アンド・マリーなら、とても好きでした。最高なフィルクを思い出します。」

「フィルク」の定義についてたずねると、ストールマンはコンセプトを説明してくれた。フィルクとは、歌詞をパロディの歌詞に置き換えたポピュラーソ ングである。(最近の数十年で、フィルカーはメロディーも書くようになった。)フィルクを書くプロセスを「フィルクする」と言って、ハッカーや SF 愛好家に人気である。古典的なフィルクには、"On Top of Old Smokey" を書き換えた "On Top of Spaghetti" や、フィルクの巨匠、変人アル・ヤンコビックがスターウォーズ指向でキンクスの「ローラ」を改作した「ヨーダ」などがある。

フォークのフィルクを聴くことに興味がありますか、とストールマンが私にたずねた。あると言うと、すぐにストールマンは意外に澄んだ声で歌い始めた。

How much wood could a woodchuck chuck,
If a woodchuck could chuck wood?
How many poles could a polak lock,
If a polak could lock poles?

How many knees could a negro grow,
If a negro could grow knees?
The answer, my dear,
is stick it in your ear.
The answer is, stick it in your ear. . .

［; 音遊びのナンセンスソング（風に吹かれて）にどなたか日本語風の歌詞を。］

歌が終わり、ストールマンは再び子供が少しだけ微笑むようにして唇をすぼめた。私は近くのテーブルを見回した。日曜日のランチを楽しんでいるアジア 系の家族連れたちは、その中に現れたあごひげのアルトにほとんど注意を払っていなかった。10 しばらくためらってから、私もついに笑った。

「最後のコーンボールは、欲しいかな。」と尋ねるストールマンの瞳はきらきら輝いていた。私がジョークを返せるよりも早く、ストールマンは、自慢げにコーンフレイクをまぶしたダンプリングを 2 本の箸でつまんで持ち上げた。「食べる資格があるのはおそらくぼくだ。」

食べるものがなくなり、会話は普通のインタビューの展開になってきた。ストールマンは椅子に寄りかかって、手で茶碗をあやしていた。私達はナップス ターと自由ソフトウェア運動の関係という話題を再開した。私は、自由ソフトウェアの原則は、たとえば音楽の出版のような類似領域に拡張すべきですか、と質 問した。

「ある問題から別の問題に答えだけを持っていこうとするのは間違ったアプローチです。」ソフトウェアプログラムと歌を対比して、ストールマンは言う。「正しいアプローチは、それぞれの著作物の類型を見てどんな結論が得られるのかを理解することなんです。」

著作権で保護された著作物ついて、ストールマンは、その世界を三つのカテゴリーに分けると言う。最初のカテゴリーには「機能的」著作物、例えば、ソ フトウェアプログラム、辞書、教科書が含まれる。第二のカテゴリーには、「証言的」と説明するのが最もよさそうな著作物、例えば、科学論文、歴史的文書が 含まれる。そういう著作物は、後続の読者や作者が勝手に書き変えると台無しになるような目的に仕えている。またそこには、個人的表現の著作物、例えば、日 記、日誌、自伝が含まれる。そういう文書を修正するのは、個人の記憶や見方を改竄することになる。それは倫理的に弁解できない行動だとストールマンは考え る。第三のカテゴリーには、アートやエンターテイメントの作品が含まれる。

三つのカテゴリーのうち、最初のものは修正版をつくる無制限の権利をユーザーに与えるべきだが、二番目と三番目は、オリジナルの作者の意思に従うよ うにその権利を調整するべきだ。しかし、コピーする自由と非営利目的で再配布する自由は、カテゴリーを問わず常に完全に残すべきだとストールマンは強調す る。もしそれがインターネットユーザーに、論説、画像、歌、または本のコピーを百部作成して、百人の他人にメールする自由を与えることを意味するなら、好 きにさせれば良い。「時に応じて個人的に再配布することは、当然、許される必要があります。警察国家だけがそれをやめさせられるわけですから。」と、ス ト

ールマンは言う。「友人との関係を裂こうとするのは反社会的です。ナップスターでは、非営利目的で楽しみのために公衆に再配布する場合でさえ、やはりそ れを許す必要があり、許されるべきだとぼくは確信しました。なぜなら、非常に多くの人がそうしたがっていて、それがとても役に立つとわかったじゃありませ んか。」

裁判所はそういう寛大な見解を認めますかね、と尋ねると、ストールマンは話の腰を折ってきた。

「それは悪い質問ですね。」と彼は言う。「というのは今、倫理の話から法解釈の話に主題をすっかり変更してしまったからです。その二つは同じ領域に あるけれども全く違う問題です。一方から他方へ飛び移るのは無益なことです。現在ある法律を裁判所がどう解釈するのかという問題は実に厳しい状況ですが、 それというのも、これらの法律はパブリッシャーたちが買ったものだからなんですね。」

コメントからストールマンの政治哲学を洞察することができる。現在、法律上の制度が会社にソフトウェア著作権を、さも土地所有権か何かのように扱う 権能を認めていても、それは、コンピュータユーザーがそれらのルールに従ってゲームしなければいけないということを意味しない。自由は倫理的問題である。 法律的問題ではない。「ぼくは現在の法律がどうなっていくべきなのか、それらがどうあるべきなのかを思い描いているんです。」とストールマンは言う。「ぼ くは法案を起草しようとしているんじゃありません。法律が何をするべきかについて考えているんです。友達とコピーを共有することを禁じる法律は、ジム・ク ロウの道徳的対応物だと思うんですよ。その法律は尊敬に値しないんだね。」

ジム・クロウの援用は、別の疑問を刺激する。ストールマンは過去の政治的リーダーからどのくらい影響を受け、どのくらいインスピレーションをひきだ しているのだろうか。1950 年代と 1960 年代の公民権運動のように、社会に変化を起こそうという彼の試みは、自由、正義、フェアプレーといった時を超 えた価値に訴えることに基礎を置いている。

ストールマンは、ひどくもつれた髪の毛を気にしながら、私のアナロジーに注意を払っていた。マーチン・ルーサー・キング牧師とストールマンを比較するところまでアナロジーを広げたとき、ストールマンは私の話をさえぎり、枝毛を切って口に入れた。

ストールマンは、もぐもぐしながら、「ぼくは彼と同じじゃないけれども、まさに同じゲームをしてるんだね。」と言った。

私は、別の比較対象としてマルコム X を示唆した。かつてのネーション・オブ・イスラムのスポークスマンのように、ストールマンは物議をかもす名声を築き、潜在的な同盟者を遠ざけ、文化的統合に対して自給自足を好むメッセージを説教している。

ストールマンは、別の枝毛を噛みながら、比較を拒否した。「ぼくのメッセージはキングのメッセージに近いんです」と彼は言う。「それは普遍的なメッ セージです。それは他人を虐待することになるある種の慣行を厳しく非難するメッセージです。それは誰に対する憎しみのメッセージでもありません。また、狭 い範囲のグループに宛てたものでもないんです。自由を尊重し、自由でいて欲しいとすべての人に促しているんです。」

ストールマンは役に立つ政治的同盟者を捨てている、と多くの人が批判し、これを心理学的に考察して、それを性格的特性として記述している人もいる。 ずいぶん宣伝された「オープンソース」という用語に対する嫌悪感の場合は、最近の提携プロジェクトへの参加の消極姿勢も理解できるように思える。この二十 年間を自由ソフトウェアを代表して歩むことに費やしてきた者として、ストールマンの政治的資本は言葉に深く投資されている。それでも、例えば、1999 年のリナックスワールドでの「ハン・ソロ」の皮肉のようなコメントは、政治の動向や市場の動向を認めたがらない不満な保守派の一人という、ソフトウェア産 業におけるストールマンの評判を補強したにすぎなかった。「私は、彼が達成したすべての仕事についてリチャードを称賛し尊敬しているのです。」と言うレッ ドハットの会長、ロバート・ヤング（Robert Young）は、ストールマンの矛盾した政治的性格を要約する。「私のただ一つの批判は、リチャードが時々友人を敵より悪く扱っているということなので す。」

［RMS:「友人」という言葉は、たとえばヤングのような人、そしてレッドハットのような会社には、部分的にだけふさわしい。それは彼らのしたこと、してい ることにも当てはまる。たとえば、レッドハットは自由ソフトウェアの開発に貢献している、それには GNU のいくつかのプログラムも含まれている。しかし、レッドハットは他のこともしていて、それは自由ソフトウェア運動の目標とは反対の作用をする。たとえば、 レッドハットの GNU/Linux のヴァージョンは非自由ソフトウェアを含んでいる。言動に転じると、全システムを Linux として言及するのは、GNU プロジェクトの友人らしくない扱い方だ。そして「自由ソフトウェア」のかわりに「オープンソース」を推進して私たちの価値を退けている。私たちが同じ方向 を目指して進んでいるときに、私はヤングとレッドハットとともに働くことができる。しかし、それは彼らをありうる同盟者とするほど多くはなかった。］

自由ソフトウェア運動が他の政治的目標を持つ運動と同盟することへのストールマンの消極性は、それに利益がないためではない。ストールマンの MIT のオフィスを訪れるなら、そこには世界中の人権侵害をカバーする左寄りのニュース記事の情報センターがある。彼のウェブサイトを訪れると、そこにはデジタ ル・ミレニアム著作権法、ドラッグ戦争、世界貿易機構に対する痛烈な非難がある。ストールマンは説明する。「自由ソフトウェア運動が他の目標を持つ運動と の同盟に参加するのは慎重にする必要があります。自由ソフトウェアの支持者の相当数がその目標に同意していないかもしれません。私たちは自由ソフトウェア 運動をいかなる政党と結びつけることも避けています。なぜなら、私たちは、支持者を追い払いたくないし、他の政党から選ばれた官僚じゃありませんから。」

彼の運動家としての傾向を知った私は、なぜストールマンはもっと大きな影響力を追求してこなかったのかと尋ねた。なぜハッカーの世界での知名度を彼 の政治的な声を増幅する演壇として使わないでいるのだろうか。［RMS: しかし、やっていますよ。- 良い機会だと思ったときにはね。それは私が stallman.org を始めた理由です。］

からまった髪から手を離し、しばらくストールマンは質問を熟考する。［ RMS:引用府付きの私の回答は質問にしっくりこない。それは別の質問にしっくりする。「なぜ信じている

他の目標ではなくて、自由ソフトウェアに焦点を合 わせているのですか。」といった質問を尋ねられていたのではあるまいか。］

「この自由の小さい水溜りの重要性を誇張するのはためらいます。」と彼は言う。「なぜなら、より良い社会と自由のために働くには、もっと良く知られ ていて、従来から存在している領域が途方もなく重要だからです。ぼくは、自由ソフトウェアがそれらと同じくらい重要だとは言わない。自由ソフトウェアはぼ くが引き受けた責任です。なぜなら、それはぼくの膝に落ちてきて、ぼくはそれについて何かができる方法がわかりました。しかし、例えば、警察の残虐行為を 止めたり、ドラッグ戦争を止めたり、まだぼくらの中にある人種差別の類を止めたり、誰でも快適な生活を送れるように助けたり、堕胎する人の権利を守ったり、自分達を神権政治から守ったりすること、これらはとても重要な問題で、ぼくのしていることより重要です。ぼくはそれらについて何かできることがわかっ たらいいのにと思うだけなんです。」

ここでも、ストールマンは彼の政治的活動を個人的確信のなせるものとして提示している。自由ソフトウェア運動の中心的教義を時間をかけて発展させ、磨きをかけてきたストールマンは、彼が支持する他の目標を彼が前進させられると信じることを躊躇する。

「それらのより大きな問題に影響を及ぼす方法がわかったらいいのにと思います。もしもそうできたら途方もなく誇りに思うでしょう。しかし、それらは 非常に困難だし、おそらくぼくよりも優れた多くの人たちがそれらに取り組んできたんです。」と彼は言う。「しかし、見たところ、これらの大きな明白な脅威 に対して他の人達は防御していたんだけれども、ぼくは防御されていない別の脅威があることがわかったんです。それで、ぼくはその脅威を防御し始めました。 それは非常に大きな脅威じゃないかもしれないけど、そこには[対抗する者が]ぼくしかいませんでした。」

最後の枝毛を噛みながら、ストールマンはお勘定にしましょうと言った。しかし、ウェイターがお金を持っていく前に、ストールマンは白いドル紙幣を抜 き出して、その上に乗せた。札は見るからに偽札で、私はそれを手にとって見ずにはいられなかった。はたして、それは偽物だった。ジョージ・ワシントンやリ ンカーンの絵のかわりに、札の正面には豚の漫画がついている。豚の上のバナーには、アメリカ合衆国（United States of America）のかわりに貪欲豚連邦（United Swines of Avarice）とあった。紙幣は零ドル札で、ウェイターがお金をとろうとしたとき、ストールマンは彼の袖をひっぱった。

唇に笑みを浮かべて、「チップに追加のゼロを加えましょう」とストールマンは言う。

理解されず、または紙幣の外観に騙されて、ウェイターは微笑んで、そそくさと行ってしまった。

「ぼくらも行ってよしってことでしょう。」とストールマンは言った。

後注

1 アンドリュー・レオナルド「自由ソフトウェアの聖者」Salon.com（August 1998）http://www.salon.com/21st/feature/1998/08/cov_31feature.html を参照。

2 レアンダー・カーニー「リナックスの忘れられた男」ワイアード・ニュース（March 5, 1999) http://www.wired.com/news/print/0,1294,18291,00.html を参照。

3「道徳的優位にあるプログラマ；自由と自由ソフトウェアを信じるリチャード・ストールマンには自由ソフトウェアは道徳の問題」、ロンドン・ガー ディアン（ 1999 年 11 月 6 日）("Programmer on moral high ground; Free software is a moral issue for Richard Stallman believes in freedom and free software," London Guardian (November 6, 1999)) を参照。 http://www.guardian.co.uk/uk/1999/nov/06/andrewbrown

これらは、宗教的な比喩をしている例を少しとりあげたに過ぎない。今のところ、いちばん極端な喩え方をしているのはリーナス・トーバルスだ。リーナ ス・トーバルスとデビッド・ダイアモンドが書いたリーナスの自伝「それがぼくには楽しかったから」（小学館プロダクション）には、「リチャード・ストール マンは自由ソフトウェアの神様だ。」とある（ 98 ページ）。

名誉ある言及をするのはラリー・レッシグだ。著書「コモンズ」（翔泳社）のストールマンの脚注でストールマンはモーゼに喩えられている。（巻末の注 ページの 2 ページ注 12 ）「... モーゼと同じく、最終的に運動を約束の地に導いたのは、OS パズルの最後の部分の開発を指導した別の指導者、リーヌス・トーヴァルズだった。またモーゼと同じく、ストールマンは運動の同志たちによって、尊敬される と同時に、毛嫌いされている。かれは現代文化のきわめて重要な側面の、厳しい、したがって多くの人にとっては頼りになる指導者だ。わたしはこの非凡な人物 の理念と献身に深い敬意を抱いているが、一方でかれの考え方に疑問を述べて、その怒りに耐えるだけの勇気を持つ人々も大いに尊敬している。」

ストールマンとの最後のインタビューで、私は宗教的比較についての彼の考えを尋ねた。「ぼくを旧約聖書の予言者に喩えたがる人がいます。その理由 は、旧約の予言者は一定の社会的慣行を間違っていると言ったからです。旧約の予言者たちはモラルの問題について妥協しませんでした。彼らは買収されず、た いてい侮辱されていたんです。」

4 レアンダー・カーニー（1999）

5 当時、ストールマンは花の学名を言っているのだとばかり思ったが、数ヶ月後、実は "rhinophytophilia" とはその仕草のこっけいな言及だったとわかった。つまり、ストールマンは鼻を花につっこんでその瞬間を楽しんでいたわけだ。- 植物との鼻のセックスの変態的実践として、それを演じながら。 ストールマンの花についての別のユーモラスな出来事はこちらへ。 http://www.stallman.org/articles/ texas.html

6「裁判所がナップスター差し止め命令の執行停止を認める」セシリー・バーンズ、スコット・アード、（2000 年 7 月 28 日）News.com http://news.cnet.com/news/0-1005-200-2376465.html を参照。

7「ナップスター事件におけるレコード産業の完勝」 RIAA press release (February 12, 2001), http://www.riaa.com/PR_story.cfm?id=372 を参照。

8「メイ・リン物語」メイ・リン・マック（1998 年 12 月 17 日） http://crackmonkey.

org/pipermail/crackmonkey/1998-December/001777.html ほかに数名の女性の情報源からこれを聞いているが、これまでのところ、この習慣について引用可能な発言として自らすすんで語っているのを私が見つけたの は、マック一人だけだ。1999 年のリナックス・ワールド・ショウで、マックは、最初それに嫌悪を表明したものの、それから不安を抑えてストールマンとダンスを踊った。

9「もしコードが自由なら、私も自由でしょ？」アンナリー・ニューヴィッツ、Salon.com（2000 年 5 月 26 日）http://www.salon.com/tech/feature/2000/05/26/free_love/print.html を参照。

10「オープンソースソフトウェア」(株式会社オライリー・ジャパン)、リチャード・ストールマン、「GNU システムと自由ソフトウェア運動」130 ページ参照。［RMS: ウィリアムズは、この寸描を私が絶望的にロマンチックで、まだ不特定の女性の気を引く努力を暗示するものとして解釈していた。MIT のハッカーは誰もこれ を信じないだろう。なぜなら、女性はほとんど気にしてくれないほど我々は幼稚だと悟っていたのだから。ましてや恋だなんて。プログラミングせいだ。我々は プログラミングに魅了されて、プログラミングをしていた。他方、これらの出来事は、そのときは完全に特定のガールフレンドがいたからこそあり得たことだっ た。もしも私がロマンチックだったとすれば、絶望的にロマンチックでも希望的にロマンチックでもなく、むしろ、一時うまく行ってロマンチックだった。

ナイーブな解釈の強みにまたがって、ウィリアムズはドン・キホーテになぞらえる所まで行ってしまった。

完全を期して、第一版から、幾分ロべたな言葉を引用しておこう。

「ロマンチックになろうなんて全然していませんよ。」と、アリックスの話を回想して、ストールマンは言う。「くすぐりかな、どちらかといえば。何を 言いたいのかというと、それはロマンチックだけど、くすぐりだったんですね。実現していれば、楽しいサプライズだったでしょう。」］

11 ストールマンの "filks" をもっと知りたい人は次を参照。http://www.stallman.org/doggerel.html ストールマンの歌う"The Free Software Song"を聞きたい人は次のサイトへ。 http://www.gnu.org/music/free-software-song.html

12 RMS: ウィリアムズは、この証券を「偽物」と呼び間違えている。それは支払いに提供できる法的証券で、いかなる債務の支払いにも零ドルの価値がある。アメリカ政府当局も零ドルの価値の金と交換してくれよう。

第六章　Emacs コミューン

1970 年代の AI ラボは、あらゆる意味で特殊な場所だった。 先端的なプロジェクトやトップクラスの研究者が、コンピュータ科学の世界で AI ラボに栄誉ある地位を与えていた。 ラボ内のハッカー文化とアナーキーなポリシーが、 反逆的で神秘的な雰囲気をかもしだしてもいた。 その後、ラボの科学者やソフトウェアのスーパースターの多くが去って行って初めて、 ハッカーたちは彼らがかつて住んでいた世界のユニークさとはかなさに気づいた。

「それはエデンの園みたいなものだったんです、」とストールマンは言い、 ラボと 1998 年のフォーブスの記事に見られるソフトウェア共有の気風を 要約する。「協力し合わない、なんてことは思いもよりませんでした。」1

こうした、いささか極端な神話めいた語り方は、ある重大な事実を強調している。 545 テック・スクウェアの 9 階は、多くの人々にとって職場以上のものだった。 ストールマンのようなハッカーたちにとって、そこは我が家だった。

「我が家」という言葉は、ストールマンの語彙の中では重い言葉だ。今でも両親を痛烈に批判するストールマンは、ハーヴァード時代に住んでいた学生寮 であるカリアー・ハウス以前に我が家があったことを認めない。また、彼はこの我が家を去るときのことを悲喜劇的な調子で話すのを常とした。あるとき、ハー ヴァード時代のことを話しているとき、ストールマンは唯一悔やまれるのは、放り出されたことだと言った。ストールマンになんで追放されるはめになったのか と聞く前に、私はストールマンの古典的な舞台設定のなかに踏み込んでいることに気がついた。

「ハーヴァードには、余りにも多くのクラスをパスした者に出て行くように求める、という方針があるんです、」とストールマンは言う。

寮を失ったが、ニューヨークに戻るつもりもなく、ストールマンは、グリーンブラット、ゴスパー、サスマンや彼以前の多くの他のハッカーたちが切り開 いた道をたどっていった。MIT の大学院生になり、ストールマンはケンブリッジの近くにアパートを借りたが、やがて AI ラボそのものを彼の事実上の我が家とみなすようになった。1986 年の講演で、ストールマンはこの頃の AI ラボの記憶を呼び戻している。

ぼくはたいていの人より多くラボで暮らしていたかもしれません、というのは 毎年とか一年置きくらいに、ぼくにはアパートがなくなって、 ラボで何ヵ月か暮らすことになったからです。 そこはとても居心地がいいんです、夏は快適で涼しいしね。 でも、ラボで寝泊りしている人を見かけるのは全然珍しいことでありませんでした。 熱くなってしまうわけです。 ハックしていられる間は帰らないんです、途中でやめたくないからです。 それで、完全に力を使い果たしたら、いちばん手近で軟らかな平面に上がっていくんです。とてもくだけた雰囲気でした。2

ラボの我が家的な雰囲気は時として問題にもなった。そこを学生寮と見る人もいたが、電子的阿片窟とみなす人もいた。1976 年の著書『コンピュータ・パワー―人工知能と人間の

理性』で、MIT の研究者ジョセフ・ワイゼンバウム（Joseph Weizenbaum）はこうした「コンピュータ浮浪者」（ワイゼンバウムは AI ラボのようなコンピュータルームにたむろするハッカーたちをこう呼んだ）をこっぴどく批判した。「彼らのしわくちゃな衣類、洗っていない頭髪、鬚を剃って いない顔、といていない髪、いずれも彼らが自分の体や自分が活動している世界に無関心なことを物語っている、」とワイゼンバウムは書いた。「[コンピュータ浮浪者たちは]およそそれに没頭しているときは、コンピュータによってまたコンピュータのためにのみ存在している。」3

出版から四半世紀も経つのに、未だにストールマンは、ワイゼンバウムが「コンピュータ浮浪者」と書いたことを聞くと腹を立て、 そのことをワイゼンバウム自身が今この部屋にいるかのように現在形で論じる。「彼は人々にただのプロになってほしいんです。 そのプロとは、できれば近寄りたくないし、さっさと忘れたいと思っているようなことを、金のためにするような人間です、」とストールマンは言う。「彼が正常な事態とみていることは、ぼくに言わせれば悲劇です。」

だが、ハッカーの生活にも悲劇がないわけではない。ストールマンは、自分が週末ハッカーから、フルタイムの AI ラボの住人になるに至った経緯を、ハッキングの幸福感でしか癒すことができなかった、一連の痛ましい不運のためだと言う。ストールマン自身が言うには、最 初の不運は、ハーヴァードを卒業したことだった。物理学の研究を続けようとして、ストールマンは MIT の大学院生になった。この大学を選んだのは自然なことだった。ウィリアム・ショックレー（36 年卒）、リチャード・ファインマン（39 年卒）、マレー・ゲ ルマン（51 年卒）といった、MIT の偉大な卒業生たちの足跡をたどるチャンスが得られるだけでなく、２ マイルほど離れたところに、AI ラボとそこの新しい PDP-10 コンピュータがあった。「ぼくの関心はプログラミングに向っていましたが、まだ両立できるんじゃないかと思っていました。」とストールマンは言う。

昼は大学院レベルの科学分野に精を出し、夜は AI ラボという僧房の中でプログラミングしながら、ストールマンは完璧なバランスをとろうとした。この優等生のシーソー生活の支点になっていたのは、フォーク ダンス・クラブ（the Folk-Dance Club）との毎週の外出だった。それは、ささやかながら異性と交流できる社交上の窓口だった。だが、MIT での１年目の終り頃、災難が見舞った。ストールマンはひざを痛めてグループを抜けなければならなくなったのだ。初め、ストールマンは怪我を一過性の問題と 見て、ダンスに行って友人と語らい、好きな音楽を聞いた。まだひざが痛み、授業が再開されようとしていた夏の終り頃、ストールマンは心配し始めた。「ひざ はちっともよくならなかったんです、」とストールマンは回想する、「すると、ダンスを完全に止めなければならなくなります。心に穴あく思いです。」

寮もダンスもなくなり、ストールマンの社交生活はしぼんでしまった。ダンスは彼が女性と出会い、時にはデートすることに成功する唯一の場だった。もうダンスができないのは充分につらいことだが、それはまた、もはやデートもできないことを意味するように思われた。

「つまり、すっかりエネルギーがなくなってしまったと感じました、」とストールマンは回想する。「ぼくには、目の前にあって一番気をひかれること以 外のことは、何をするエネルギーもなくなっていました。何か他のことをするエネルギーは失せました。ぼくは、完全に落ち込んでいました。」

ストールマンはさらに世界から身を退き、AI ラボの仕事に全面的に集中した。1975 年 10 月、彼は MIT を退学し、物理から手を引いた。二度と研究に戻る気はなかった。かつて趣味だったソフトウェアのハッキングが、天職になった。

このころを振り返り、ストールマンは、フルタイムの学生がフルタイムのハッカーになったのは必然的だったと見る。遅かれ早かれ、コンピュータハッキ ングというサイレンの呼び声は、他の専門性を追求することへの関心を圧倒しただろう、と彼は信じている。「物理や数学では、ぼくはどうしたら貢献できるか 分かりませんでした、」と、ひざを痛める前の奮闘ぶりを回想して、ストールマンは言う。「これらの分野のいずれかで自慢できるほど進歩することもできたか もしれませんが、どうすればいいのかは皆目分かりませんでした。どこから始めればいいかも知りませんでした。ソフトウェアなら、走らせて役に立つようなも のを書く正しい方法が分かりました。それが分かっていることが嬉しくて、それをもっとやりたいと思うようになりました。」

ハッキングを快楽と同一視したのはストールマンが最初ではなかった。AI ラボを構成していたハッカーの多くが、同様な、達成されなかった学問的な経歴を自慢にしていた。多くが、数学や電子工学の学位を求めてやって来たが、学問 的な経歴や専門への野心を、今まで誰も取り組んだことのない問題を解決することの興奮と引き換えにした。神学大全の執筆に長時間取り組んだために、ときど き霊視に到ったとされる聖トマス・アキナスのように、ハッカーたちは、ひたすら精神を集中し肉体を消耗させることで、超越的な内部状態に達した。 ストールマンはドラッグに手を出さなかったが、多くのハッカーと同様、彼も 20 時間に及ぶコーディング祭りの終りごろに訪れる「ハイ」な状態を楽しんだ。

だが、おそらく一番喜ばしい感情は、個人的な達成感だった。ハッキングにかけては、ストールマンは申し子だった。子供のころから真夜中に勉強する習 慣があったので、彼は少ししか寝ずに長時間働くことができた。10 歳の頃から社会から締め出されていたので、独りで働くことには何の困難もなかった。そして、生まれつき論理や洞察に長けている数学者のように、ストールマ ンには、多くのハッカーたちが時間を空費していた設計上の障害を回避する能力があった。

「彼は特別でした、」と AI ラボの教員で、1985 年以来、自由ソフトウェア財団の役員をしている、ジェラルド・サスマン（Gerald Sussman）は回想する。ストールマンを「明解な思索家で明解な設計者」と評するサスマンは、ストールマンを 1973 年と 1975 年の AI 研究プロジェクトに誘った。どちらも、人間のエンジニアがするような仕方で回路図を解析できる AI プログラムを作成することを目指していた。プロジェクトには、同じ課題に人間がどうアプローチするかについての（サスマンが供給する）理解はもちろん、特 に

AI アプリケーションのために作られたプログラミング言語、Lisp（リスプ）の専門的コマンドが必要だった。1975 年のプロジェクトは、依存に基づく後戻り（dependency-directed backtracking）または真理の維持（truth maintenance）と呼ばれる AI 技術を開拓した。それは一時的な仮定を事実と仮定することで構成されていた。もしも、それらが矛盾に導けば感知（noticing）し、もしも、それが生 じれば、適切な前提を再考する。

これらのような公式のプロジェクトで働いていないとき、ストールマンは自分の時間をいくつかのペット・プロジェクトに当てた。それはラボのソフト ウェア基盤を改善しようというハッカーの最良の関心事から発したもので、この時期のストールマンの最大のペット・プロジェクトの一つが、ラボのエディタ・ プログラム TECO（ティーコ）だった。

1970 年代のストールマンの TECO に関する仕事の話は、後にストールマンが自由ソフトウェア運動で発揮するリーダーシップと密接に結びついている。またそれはコンピュータの進化の歴史の中 で重要な一段階でもあるので、この進化を簡単に要約しておく必要がある。1950 年代から 1960 年代にかけて、コンピュータが初めて大学内に登場したとき、コンピュータのプログラミングは信じ難いほど抽象的な営みだった。マシンと対話するのに、プロ グラマは一連のパンチカードを作った。各カードは、個々のソフトウェアコマンドを表していた。 次にプログラマがカードを中央システムの管理者に渡すと、管理者はカードを 1 枚ずつマシンに挿入し、マシンが新たに一組のパンチカードを吐き出すのを待った。プログラマはこのカードを出力として解読するのであった。「バッチ処理」として知られるこのプロセスは煩雑で、時間を要した。それはまた権威の乱用を 助長した。初期のシステム管理者が、どのジョブに優先権を与えるかを指示する権限を持っていたことが、ハッカーが集権化を生来毛嫌いする原因の一つだっ た。

1962 年、AI ラボの前身である MIT のプロジェクト、MAC に関わっていたコンピュータ科学者とハッカーたちは、この不満を解消するための一歩を踏み出した。当初は、「時間どろぼう」として知られていた、タイム シェアリング（時分割）は、複数のプログラムがマシンの動作能力を利用することを可能にした。テレタイプをインターフェースにすることで、一連のパンチ穴 ではなく、実際の文字によってマシンと対話できるようになった。プログラマはコマンドをタイプし、マシンが生成する出力を 1 行ずつ読んだ。

1960 年代の終わり頃、インターフェースの設計がさらに飛躍した。スタンフォード研究所に勤務していたダグ・エンゲルバートは、有名な 1968 年の講義で、モダンなグラフィカルインターフェースのプロトタイプをお披露目した。コンピュータにテレビをとりつけ、エンゲルバートが「マウス」とあだな をつけた位置指示装置をつけ加えることで、この科学者は MIT で開発されたタイムシェアリングシステムよりずっと対話的なシステムを創造した。ビデオディスプレイを高速プリンタのように使うことで、エンゲルバートの システムは、ユーザが画面の中でカーソルを動かし、カーソルの位置がコンピュータによりリアルタイムに更新される様を見られるようにした。ユーザーは突 然、画面のどこにでも文章を配置できるようになった。

こうした技術革新が商品市場に現れるまでには、さらに 20 年を要した。だが、1970 年代には、表示端末としてはビデオ画面がテレタイプにとって代り始め、行単位ではなく、フルスクリーンで編集する機能への潜在能力を作りだした。

このフルスクリーンの機能を利用した最初のプログラムの一つが、MIT AI ラボの TECO だった。TECO は Text Editor and COrrector（テキスト編集・修正プログラム）の省略表記であり、ハッカーたちが、ラボの PDP-6 マシンのために、旧式のテレタイプ用行エディタの機能を向上させたプログラムだった。4

TECO は旧式のエディタよりかなり改善されていたが、まだ欠点があった。文書を作成して編集するのに、プログラマは 1 回の編集を指定する毎に、一連のソフトウェアコマンドを入力しなければならなかった。これは抽象的な手続きだった。キー入力の度に文章を更新していく現在 のワープロとは異なり、文章を変更するだけでも、TECO は、ユーザに拡張された一連の編集コマンドと「コマンドの終り」を表す一連の文字を入力することを要求した。しばらくすると、ハッカーは文書全体を編集 モードで書くまでに習熟していったが、後にストールマン自身が指摘したように、この手順は「目隠ししてチェスをするのに匹敵する精神的技量」を要した。5 この手順を軽減するために、AI ラボのハッカーたちは、「原文」モードと「表示」モードの両方を別の画面に表示するシステムを造った。しかし、こうした革新的なハックによっても、モード 間の切り替えは煩わしいものだった。

TECO はこの頃コンピュータの世界に出回っていた唯一のフルスクリーンエディタではなかった。1976 年にスタンフォード人工知能研究所を訪れたとき、ストールマンは E という編集プログラムに出会った。このプログラムにはある内部機能があって、それによってユーザは各コマンドをキー入力した後で表示される文章を更新する ことができた。1970 年代のプログラミングの言葉で言えば、E は WYSIWYG エディタのはしりだった。「見たとおりのものが手に入る（what you see is what you get）」の省略表記である WYSIWYG は、背後にいるエディタプログラムによって作業をするのではなく、ユーザが表示されている文章の中を移動することでファイルを操作できることを意味した。 6

このハックに影響されたストールマンは、MIT に帰ると、TECO の機能を同様な仕方で拡張する方法を探した。彼はカール・ミケルソン（Carl Mikkelson）が書き、それを起動する二つのキー入力の組合せから命名した、Control-R という TECO の機能を見つけた。ミケルソンのハックは、TECO を通常の抽象的なコマンド実行モードから、より直感的なキー入力ベースのモードに切り替えるものだった。ストールマンはこの機能を繊細かつ大胆に改造し た。 彼は他の TECO のコマンド文字列つまり「マクロ」も他の二つのキーの組合せで起動できるようにした。今までユーザがコマンド文字列を入力し、入力した後で捨てていたの を、ストールマンのハックは、マクロの技をファイルに保存し、意のままに呼出せるようにした。ミケルソンのハックは、TECO を WYSIWYG エディタに昇格させた。「これが真の突破口でした、」とその当時の同僚で AI ラボのハッカーだったガイ・スティール（Guy Steele）は言う。7

ストールマン自身の記憶によれば、マクロのハックはさらなる発明の爆発を引き起した。「誰も彼もが自分用に再定義したスクリーンエディタコマンドのコレクションを書いていました。自分がよく行いそうなこと全てに対するコマンドです、」とストールマンは後に回想している。「みんなはそれらを回覧して、改良し、より強力に、より汎用的にしていきました。 再定義されたコマンドの集まりは、しだいに一人前のシステムプログラムになっていきました。」8

非常に多くの人々がマクロというイノベーションの便利さに気づき、それを自分のTECO プログラムに取り込んでいったために、TECO エディタはマクロに霊感を受けたマクロマニアには二次的なものになった。「ぼくらは心の中では、それをエディタではなく、プログラミング言語に分類し始め ていました、」とストールマンは言う。ユーザーはこのソフトウェアをいじったり、新しいアイディアを交換することに快感を覚えていた。9

この爆発から 2 年後、発明の勢いが危険な副作用を生み始めた。この爆発的な成長は、ハッカーの協調的アプローチのわくわくするような有効性を示したが、過度の複雑さをもたらすようにもなった。「バベルの塔効果が現れたんです、」とガイ・スティールは言う。

この効果はそれを生み出した精神を殺そうとしていた、とスティールは言う。ハッカーがITS を設計したのは、プログラマが知識を共有し、お互いの仕事を改善することを容易に行えるようにするためだった。それは、他のプログラマのデスクに座り、プ ログラマの仕事を開いて、彼のソフトウェアの中に直接コメントや修正を行うことを意味した。「誰かに何かのプログラムやデバッグの仕方を教えるとき、端末 の前に座って、彼らの代りにそれをやって見せるのが一番簡単な場合がよくあります、」とスティールは説明する。

マクロ機能は、2 年目以降、この長所を損い始めた。この新しいフルスクリーン機能を熱心に取り込もうとしたハッカーたちは、TECO を自分版にカスタマイズしていったために、他のハッカーの端末の前に座ったハッカーは、どのマクロコマンドが何をするのかを理解するのにいつも最初の 1 時間を費すようになった。

これにいらだったスティールは、自分でこの問題を解決しようとした。4 つの異なるマクロパッケージを入手し、最も便利なマクロコマンドを文書化した図を作り始めた。図に指定した設計を実装していく内に、スティールの言によれば、ストールマンの注意を引いた。

「彼はぼくの肩越しにのぞいてきて、何してるの、と聞きました、」とスティールは回想する。

ストールマンと余り交流のなかった穏やかなタイプのハッカーのスティールにとって、この記憶は今でも突出している。他のハッカーの肩越しに仕事をの ぞき見ることは、AI ラボではふつうのことだった。ラボでの TECO のメンテナであるストールマンは、スティールの仕事を「面白い」と思い、それを仕上げようと直ちに乗り出した。

このプロジェクトの新しい名前、Emacs は、ストールマンに負っている。Emacs は、「編集用マクロ（editing macros)」の省略記法だが、これは 2 年前にマクロが爆発したときに起った進化上の卓越性を含意していた。そしてそれは、ソフトウェアプログラムの名前辞書

の隙間を利用したものでもあった。 ITS のプログラムに文字 "E" で始まるものが存在しないことに気づいて、ストールマンは Emacs を選んだ。こうすると、プログラムを 1 文字で参照できるのだ。またしても、ハッカーの効率性への情熱が、その痕跡を残したことになる。10

もちろん、全員が Emacs に乗り換えたわけではなかった、または直ぐに乗り換えたわけではなかった。これまで通り、ユーザーは、TECO ベースのエディタを手入れして走らせ続けることができた。しかし、Emacs は、ある部分を他人が変更しないで使っているときにも、置き換えたり追加したりすることがとても簡単にできる設計だったので、たいていは、Emacs に乗り換えたほうが良いことに気付いた。

「一方で、ぼくたちはもう一度、一様性のあるコマンドセットを作ろうとしていましたが、他方では、これを開かれたものにしておきたかったんです。なぜなら、プログラム可能であることは重要なことだったからです、」とスティールは回想する。

するとストールマンは別の問題に直面するようになった。ユーザたちが何かを変えたが、その変更をコミュニティの他のユーザに伝えなかったら、バベル の塔効果が他のところで再現するだけだろう。技術革新の共有というハッカーの教義に回帰するべく、ストールマンは利用上の規約を定めた一文をソースコード に埋め込んだ。行った拡張については全てを送り返すという条件で、ユーザは自由にコードの変更や再配布をしてよい、というものだ。ストールマンはそれを「Emacs コミューンへの参加」と名づけた。TECO がただのエディタ以上のものになったように、Emacs はただのソフトウェアプログラム以上のものになった。ストールマンにとって、それは社会契約だった。このプロジェクトを文書化した 1981 年の覚書に、ストールマンはこの契約条項をはっきり書き記していた。「Emacs は、」と彼は書いた、「コミューン的な共有を基盤として配布される。それは全ての改良を取り込んだ上で再配布されるために、全ての改良が私に送り返される ことを意味する。」11

オリジナルの Emacs は、PDP-10 コンピュータ上でのみ動いたが、まもなく他のコンピュータのユーザーも Emacs で編集することを欲した。爆発的イノベーションが十年間続き、数多くの Emacs 風プログラムが生まれたが互換性はまちまちだった。Emacs コミューンのルールはそれらに適用されなかった。それらのコードは分離されていたからだ。その幾つかは、ストールマンのオリジナルの Emacs との関係をユーモラスな再帰的名称で引用した。Sine（Sine is not Emacs; Sine は Emacs に非ず）、Eine（Eine is not Emacs; Eine は Emacs に非ず）、Zwei（Zwei was Eine initially; Zwei は始め Eine だった）。オリジナルがそうだったように、本物の Emacs はユーザーのプログラミング可能性を提供しないといけなかったが、それが欠けているものは「作り物の」Emacs と呼ばれた。例えば、Mince（Mince is Not Complete Emacs; Mince は完全な Emacs に非ず）がそれだ。

AI ラボではストールマンが Emacs を開発している一方で、他の所では、ハッカーコミュニティを不安にさせる開発があった。1979 年にブライアン・リードが スクライブに「時限爆弾」を仕込み、ユニロジック社がソフトウェアへの未払いユーザーのアクセスを制限す

ることを可能にしたのは、ストールマンにとって悪 い前兆だった。「彼はそれを生涯で目撃した最もナチス的な出来事だと考えていました、」とリードは回想する。リードはユーズネットの alt 階層の共同創始者として、後日インターネットで名声を博することになったが、それでも、彼は、1979 年にしたことの償いは未だ終っていない、少くともストールマンの目から見れば、と言う。「彼は全てのソフトウェアは自由でなければならない、また、ソフト ウェアに課金するような企ては人類に対する犯罪だ、と言っていました。」12

ストールマンにはリードの取引きを食い止める力はなかったが、ハッカー精神に悖ると思われる他の行動形態を抑えることはできた。Emacs「コ ミューン」の中心的なコードメンテナとして、ストールマンはその権力を振るって政治的効果をもたらそうとし始めた。パスワードをめぐるコンピュータ科学研 究所の管理者とのいざこざの最終段階で、ストールマンはソフトウェア「ストライキ」を開始し、研究所のコンピュータのセキュリティシステムを拒否するま で、研究所のメンバーに Emacs の最新版を送付するのを断った。13 これは制裁措置というより、ジェスチャーだった。なぜなら、彼らが自分でインストールするのを止めさせるものは何もなかったのだから。しかし、真意は伝 わった。ITS システムにパスワードをつけることは、非難と反発をもたらすのだ。

「多くの人がぼくに腹を立てていました、 ぼくが彼らを人質にとっているとか、彼らをゆすっている、とか言ってね。 ある意味では、ぼくはそうしていました、」 後日、ストールマンは作家のスティーブン・レヴィに語っている。 「彼らが広範な人々に対して狼藉を働いていると考えて、ぼくも彼らに対して狼藉を働きました。」14

しだいに、Emacs はハッカー倫理を売り込むための道具となった。ストールマンがソフトウェアに埋め込んだ柔軟性は、協調を促進するだけでなく、それを要求した。Emacs の最新の進化についていっていなかったり、自分の貢献をストールマンに送り返していないユーザには、最新のブレークスルーを見逃すリスクがあった。このブ レークスルーは多岐に渡っていた。20 年後には、ユーザは Emacs を極めて多様な用途のために改造したので -- スプレッドシート、電卓、データベース、ウェブブラウザのために利用した -- その後の Emacs の開発者たちは、その融通無碍な機能性を表すのに、溢れかえるシンク(overflowing sink)、と呼んだ。 「これこそぼくらが伝えたかったアイディアなんです、」とストールマンは言う。 「それが内に秘めていた素材の多さは驚異的で、恐しいほどだったんです。」

AI ラボのストールマンの同期たちは、さらに寛大だ。1970 年代にストールマンと仕事をした MIT の大学院生で、後に自由ソフトウェア財団の設立委員としてストールマンを補佐したハル・エイベルソン(Hal Abelson)は、Emacs を「間違いなく絶品」だと言う。プログラマに、システムを台無しにせずに、新しいソフトウェアライブラリや機能を追加する手段を与えたことで、ストールマ ンは将来の大規模で協調的なソフトウェアプロジェクトへの道を整えたのだ、とエイベルソンは言う。「その構造が堅固なので、ゆるく結びついた世界中の人々 が、それに貢献できるんです、」とエイベルソンは言う。「私の知る限りでは、それ以前にこんなことはなかったと思います。」15

ガイ・スティールも同様の賞賛を表明する。現在サン・マイクロシステムズの研究者である彼は、ストールマンを、なによりも「バグの少ないコードを大 量に生み出す能力がある最高のプログラマ」として記憶している。性格は完全に噛み合ったわけではなかったが、スティールとストールマンは、スティールがス トールマンの強烈なコーディングスタイルを垣間見られるぐらいの期間は協働したことがあった。彼は 1970 年代の終り頃の有名なエピソードを思い出す。そのとき、この二人のプログラマは協力して、エディタの「整形」機能を書いた。当初のスティールの想定では、 整形はキー入力で起動する別機能で、Emacs のソースコードを整形し、読み易く、場所もとらないようにし、さらにプログラムの WYSISIG 度を向上させるものだった。この機能は戦略的なものだったので、ストールマンの強い関心を引き、スティールがそれを書き上げる前から、彼とストールマンは その改良版を計画していた。

「ある朝、ぼくらは席につきました、」とスティールは回想する。「ぼくがキーボードの前にいて、彼はぼくの肘のところにいました、」とスティールは言う。「彼は間違いなくぼくにタイプさせるつもりでしたが、タイプする内容にも口を出しました。」

このプログラミングセッションは、10 時間に及んだ。この間ずっと、とスティールは言う、彼もストールマンも、休憩も、おしゃべりもしなかった。セッションの終る頃には、彼らはこの整形用ソー スコードをハックして 100 行以下にまで切り詰めていた。「ぼくの指はずっとキーボードの上にありました、」とスティールは回想する、「でも、二人のアイディアが画面に流れ込んでい くような気がしていました。彼がタイプする中身を言い、ぼくがそれをタイプしました。」

このセッションの長さには、スティールは AI ラボから帰ろうとするときまで気づかなかった。 545 テック・スクウェアのビルの外に出たとき、彼は辺りが真暗になっているのに驚いた。 プログラマとして、スティールは長時間のコーディングセッションには慣れていた。 だが、今回のセッションでは何かが違った。 ストールマンとの作業はスティールに、全ての外的な刺激を排除して、手元の課題に全ての精神的なエネルギーを集中することを強制した。 振り返って、スティールが言うには、ストールマンの心には、心を高揚させると同時に怖気づかせるものが混ざり合っていることに気づいた。 「この後、まず思ったのは、これがすごい経験、とても強烈な経験だったということでした。そして、もう二度とごめんだ、ということでした。」

後注

1 ジョッシュ・マクヒュー「ハッキングが好きだから」『フォーブス』誌（1998 年 8 月 10 日号）を参照。http: //www.forbes.com/forbes/1998/0810/6203094a.html

2 ストールマン（1986 年）参照。

3 Joseph Weizenbaum, Computer Power and Human Reason: From Judgment to Calculation (W. H. Freeman, 1976) 116 ページを参照。 （日本語訳：ジョセフ・ワイゼンバウム『コンピュータ・パワー—人工知能と人間の理性』、サイマル出版会、1979 年、秋葉

忠利訳）

4. ジャーゴンファイルによれば、TECO という名前は当初テープ編集・修正プログラム（Tape Editor and Corrector）を表していた。［「テープ」とは当時の入出力装置である「紙テープ」のこと。訳者］

5. リチャード・ストールマン「EMACS: 拡張可能、カスタム化可能なディスプレー・エディター」AI ラボ・メモ（1979 年）を参照。 私の引用は、このメモを更新した HTML 版からのもので、以下で参照できる。

http://www.gnu.org/software/emacs/emacs-paper.html

6. リチャード・ストールマン「Emacs フルスクリーンエディター」（1987 年）を参照。

http://www.lysator.liu.se/history/garb/txt/87-1-emacs.txt

7 同書

8 同書

9 同書

10 同書

11 ストールマン（1979 年）：#SEC34 を参照。[注 6 で参照しているメモの脚注に、この配布に関する文章が見られる。訳者]

12 1996 年のオンライン・マガジン『MEME』誌とのインタビューで、 ストールマンは、Scribe 売却の件にはうんざりだと漏らしたが、 リードの名前には触れようとしなかった。「問題は、誰もこの学生の行為を咎めたり、罰しようとしなかったことです、」とストールマンは言った。 「その結果、他の人々も彼にならおうという気にさせられたんです。」『MEME』誌 2.04 を参照。

http://memex.org/meme2-04.html

13 Steven Levy, Hackers（Penguin USA [ペーパーバック]、1984）149 ページを参照。（日本語訳：スティーブン・レビー『ハッカーズ』、工学社、1987 年、松田信子、古橋芳恵訳）

14 同書

15 この章を書くに当たって、私はソフトウェアとしての Emacs の意義より、 その社会的な意義に焦点を当てることにした。 そのソフトウェアとしての側面についてもっと読みたいという向きには、 ストールマンの 1979 年のメモをお勧めする。 特に、「インストールされるツールの開発による研究」という節(#SEC27)がお勧めだ。 この節は、技術的な知識のない読者でも近寄ることができ、 ストールマンの政治哲学とソフトウェアの設計哲学がいかにからみ合っているかを明らかにしてくれる。 以下に一部引用しておく。

慎重な設計という手順をたどるのでは EMACS には到達できなかっただろう。 なぜなら、こういう手順では始めから見えていて、 何が望ましいのかについても、始めからきっちり決められているようなゴールにしか到達できないからだ。 私がそれをつくり出すまで、私を含め誰も拡張可能なエディターなど夢想だにしていなかったし、 それを体験するまで

その価値を評価することもなかった。 今 EMACS が存在するのは、到達点の見えない道をたどる内に、 私が個人的に便利だと思う小さな改良を遠慮無く行っていったことによるのだ。

第七章　率直な道徳的選択

1983 年 9 月 27 日、Usenet のニュースグループ、net.unix-wizards に接続したコンピュータ・プログラマは、いつもと違ったメッセージに遭遇した。深夜、正確には午前 12 時 30 分に投稿され、rms@mit-oz と署名されたメッセージの件名は、そっけないが人目を引いた。「新しい UNIX の実装」とある。だが、メッセージに Unix（ユニックス）の新ヴァージョンの紹介はなく、メッセージの最初のパラグラフが動員の呼びかけで始まっていた。

今年の感謝祭から、私は Unix と完全に互換性のあるソフトウェアシステム、GNU（Gnu's Not Unix; Gnu は Unix に非ず）を書き始めます。 それを使える人には誰にでも無料で差し上げます。時間とお金、プログラムと機材での貢献が大いに必要です。1

経験を積んだ Unix の開発者にとって、このメッセージは理想主義と自信過剰を混ぜ合わせたものだった。著者はすでに成熟した Unix オペレーティング・システムを土台から再構築すると誓い、そのうえ、あちこちを改良することまで提案している。著者が予測する新しい GNU システムは、テキストエディタや Unix 互換アプリケーションを走らせるシェルプログラム、コンパイラ、「そしてその他いくつか」の一般的コンポーネントすべてを持っている。2 さらに、プログラミング言語 Lisp をベースにしたグラフィック・ユーザー・インターフェイスやクラッシュ耐性化されたファイルシステム、それに MIT のインターナル・ネットワーキング・システムに基づいたネットワーキング・プロトコルのような、まだ Unix システムが提供していない魅力的な機能がたくさん含まれるという。

「GNU は Unix プログラムを実行できるようになるでしょうが、全く同じものになるわけではありません」と著者は書く。「他のオペレーティング・システムの経験を基礎にして、便利なものをすべて改良するつもりです。」

読者側に懐疑的な反応があることを見越して、著者はオペレーティング・システムの概要を補足する「私は誰か」と題した略歴を加えた。

私はリチャード・ストールマンです。たくさん模倣されたオリジナル EMACS エディタの考案者で、今は MIT の人工知能研究所にいます。コンパイラ、エディタ、デバッガ、コマンドインタプリタ、非互換時分割システム（ITS）や Lisp マシン・オペレーティング・システムについて広く取り組んできました。ITS 端末の独立ディスプレイのサポートを開拓し、Lisp マシンにクラッシュ耐性ファイルシステムを一つ、ウィンドウシステムを二つ実装しました。3

運命により、ストールマンの夢のような GNU プロジェクトは、感謝祭の開始日よりも遅れた。しかし、1984 年の 1 月までに、ストールマンは約束を果たし、彼は Unix のソフトウェア開発の世界に完全に没頭していった。ITS 育ちのソフトウェア設計者にとって、それはまるでムーア様式の宮殿の代わりに郊外型ショッピングモールを設計するようなものだった。しかし、それでも、 Unix 的オペレーティング・システムを作ることには隠れた利点があった。ITS は、強力だったがアキレス腱があった。MIT のハッカーは、DEC 社

が製造した PDP 系マシンに固有の利点を生かして設計をした。しかし、AI ラボの管理者が 1980 年代初頭に研究所の強力な PDP-10 を段階的に廃止する選択をしたとき、かつてハッカーが活気ある都市に例えたオペレーティング・システムは瞬く間にゴーストタウンになってしまった。一方、 Unix は、移植性を目指して設計され、そのような危険に対する免疫を持っていた。最初に AT&T の下級エンジニアたちによって開発されたプログラムは、企業経営者のレーダーから抜け出し、資金繰りの厳しい大学のコンピュータシステムに安住の地を見つ けた。Unix の開発者たちは、MIT の同胞たちよりも小さなリソースを使いながら、雑多な寄せ集めのハードウェアシステムに、すなわち、大半の AI ラボのハッカーからは小さなタスクに合うだけだと思われていた 16 ビットの PDP-11 から VAX 11/780 のような 32 ビットのメインフレームに至るすべてに、ソフトウェアを載せるべくソフトウェアをカスタマイズした。1983 年までに、最も有名なサンマイクロシステムズを含むいくつかの企業は「ワークステーション」と名付けたデスクトップ・コンピュータのさらに強力な世代を開 発していた。それは、より古い世代の PDP10 の力に匹敵するパワーを持ったマシン上でますます偏在化するオペレーティングシステムの利点を生かしたものだった。

この移植性を円滑にするために、Unix の設計に従事していた開発者たちは、ソフトウェアとマシンの間に必ずもう一つ追加的抽象レイヤーを置くようにしていた。オペレーティングシステムを、AI ラボのハッカーが ITS や PDP-10 でしたように、特定のマシンのタイプの命令で書かないで、Unix の開発者は、C と呼ばれる高級言語で書いた。インターフェイスと仕様書が噛み合うことに焦点を合わせて、彼らはどんなマシンでも走るようにすぐに修正できるシステムを書 いた。実際のコンポーネント自体よりも、むしろオペレーティング・システムの多くのサブコンポーネントをまとめるものに焦点を合わせた。もし、ユーザーが あるコンポーネントを気に入らなければ、インターフェイスの仕様書は、個別のサブコンポーネントを抜き出し、それを修繕し、あるいは置き換え、もっと良い 何かにすることができた。簡単に言うと、Unix のアプローチは、柔軟性と経済性を促進するものだったため、急速に導入されていった。4

ストールマンが GNU システムの開発を始めようと決心したきっかけは、AI ラボのハッカーたちが長い間育ててきた ITS システムの廃止だった。ITS と AI ラボのハッカーコミュニティの死は、ストールマンにとってトラウマ的な打撃だった。ゼロックスのレーザープリンタのエピソードが占有的ソフトウェアの不正 義を彼に認識させたとすれば、コミュニティの死は彼に占有的ソフトウェアに降伏するかそれに反対するかの選択を迫った。

ITS を構成したソフトウェアコードがそうであるように、ITS 廃止の起源は、それ以前の時期に遡ることができる。1980 年までに、ラボの大半のハッカーは、Lisp マシンとそのオペレーティング・システムの開発に従事していた。

1950 年代後期の MIT の人工知能研究者の一人で、人工知能研究のパイオニアであるジョン・マッカーシー（John McCarthy?） がつくった Lisp（リスプ）はエレガントな言語

で、不規則な構造のデータを操作する複雑なプログラムを書くのに適している。この言語の名前は LISt Processing（リスト・プロセッシング）を略したものだ。マッカーシーがスタンフォード大学人工知能研究所へ転任するのに伴い、MIT のハッカーたちはこの言語を磨き上げて MACLISP という名の方言を作り出した。MAC とは、DARPA より資金提供され、AI ラボとコンピュータ・サイエンス研究所を創設することになった研究プロジェクト、プロジェクト MAC のことを指している。AI ラボの大物ハッカー、リチャード・グリーンブラットを先頭に、1970 年代後期の AI ラボのプログラマたちは、Lisp が効率的で都合良く走ることに特化したコンピュータ、Lisp マシンを設計し、そのための完全に Lisp ベースのオペレーティング・システムを開発した。

1980 年までに、ハッカーの二つのライバルグループが、Lisp マシンのコピーを製造販売する二つの会社を設立した。グリーンブラッドは、Lisp マシン社（Lisp Machines Incorporated）を始めた。彼は外部からの投資を避けて「ハッカーの会社」を作ることを計画していた。大部分のハッカーは、従来型のスタート アップであるシンボリクス社（Symbolics）に参加した。1982 年には、彼らは MIT で働くことをすっかりやめてしまった。

専念してくれるハッカーがほとんどいなければ、プログラムやマシンの修復にはもっと時間がかかる、つまり、それはちっとも修復されないということ だった。ストールマンによれば、さらに悪いことにラボの「人口構造の変化」が起きた。かつて AI ラボで発言力のある少数派だったハッカーはその地位を失い、その一方で、「(PDP-10 を）心から愛していない教授や学生が今までにないくらいの多数派 になってしまった。」5

1982 年、AI ラボは 12 歳を過ぎていたメイン・コンピュータの PDP-10 を交代させた。デジタル社（Digital）の最新モデル、デックシステム 20 は、ユーザーのプログラムの互換性を持っていたが、もしハッカーが同じオペレーティングシステムで走らせ続けたければ、思い切った書き直し、つまり、 ITS の「移植」が必要だった。恐ろしいことに、研究所は所内でプログラミングの才能のある人間の必要数を既に失っていて、そのため、AI ラボの教授陣はデジタル社の開発したオペレーティング・システム、Twenex を強要してきた。多勢に無勢、ハッカーには承諾する選択肢しかなかった。

「『システムを管理するハッカーがいなければ大惨事になる。商用ソフトウェアを使うべきだ』と（教授陣は）言うのです。」ストールマンは数年後にこ う振り返っている。「『会社がシステムを保守してくれることを期待できる』と言っていました。それで彼らは完全に間違っていると分かったけど、商用ソフト ウェアを使うことになりました。」6

最初、ハッカーたちは Twenex システムも打倒されたがっている、もう一つの権威のシンボルと見ていた。システムの名前自体が異議の申し立てだった。公式には DEC が TOPS-20 と名付けていたが、それは DEC が PDP-10 用に販売していた商用オペレーティングシステム、TOPS-10 の後継だった。ボルト・ベラネク・ニューマンは TOPS-20 の元になったその改良版を開発して、それを Tenex と名付けていた。7 Twenex という呼び

名を考案したハッカーのストールマンは、TOPS-20 という名前を使わないで済むようにその呼び名を思いついたのだという。「システムはトップと呼ぶにはほど遠かったから、そんな名前で呼ぶしかなかったんで す」とストールマンは回想している。「それで、Tenex という名前に『w』を一つくっつけて、Twenex と呼ぶことにしました。」

Twenex/TOPS-20 システムを動かしているマシンにも、冷笑的な意味の『オズ』というあだ名があった。ハッカーの伝説の一つによれば、マシンにそのあだ名が付いたのは、端末に電力を供給するのにより小さい PDP-11 が一台必要だったからだ。あるハッカーが、KL-10 と PDP-11 のセットアップを初めて見たときに、それを映画『オズの魔法使い』の中の大げさな魔法使いの登場場面になぞらえた。「我こそは偉くて強いオズ」、そのハッ カーは歌った。「コンソールのうしろの PDP-11 を気にすることはない。」8

最初に KL-10 に遭遇したハッカーが笑ったとしても、Twenex と遭遇したときに、笑いはすぐに消えてしまっただろう。Twenex は備えているセキュリティを強化したばかりか、システムのソフトウェア・エンジニアは、セキュリティに配慮してツールやアプリケーションを設計していた。 コンピュータ・サイエンス研究所のセキュリティシステムとのパスワードを巡るいたちごっこが、今やシステム管理を巡る徹底抗戦になった。システム管理者は、セキュリティ対策を施さなければオズのシステムはより頻繁に不測のクラッシュに見舞われると主張していた。ハッカーたちは、クラッシュはソースコード を徹底的に見直すことで回避できると主張した。しかし、残念ながらそのような総点検を実行する時間とやる気のあるハッカーの数はシステム管理者の主張が 通ってしまうほど減少していた。

当初の政策は、ラボのメンバーであれば誰でもセキュリティ制限をバイパスする「実力者特権」を持つことができるというものだった。しかし、「実力者 特権」を持つ者は他の者のそれを奪うことができた。奪われた者はその特権を復活できない。この状況は、ある小グループのハッカーに、彼らを除く全員の「実 力者特権」を取り消すことによる完全なコントロールを掌握する誘惑を与えた。

パスワードを入手し、起動中にデバッガーを適用し、ストールマンはこれらの企てをうまく失敗させていた。二度目の「クーデター」を失敗させた後、ストールマンは AI ラボの全員に対して警報を発した。9

「また権力を握ろうとする試みが行われた」とストールマンは書く。「これまでのところ、貴族的勢力は敗北している。」アイデンティティを守るため、ストールマンはメッセージに「自由オズ放送」と署名した。

変装は、せいぜい少しやってみましたといった程度のものだ。1982 年までにストールマンのパスワードと秘密主義嫌いは広く知れ渡っていたため、AI ラボの外にいたユーザーは ARPAnet を利用するのに彼のアカウントを踏み台として利用していた。ARPAnet は、研究助成金を支給されて今日のインターネットの基盤となったネットワークである。1980 年代始めのこの手の「観光客」の一人で、ハッカーの情報網を通じてどの部外者も MIT ご自慢の ITS システムにアクセスするには RMS というイニシャルと、システムからパスワ

ードが要求されれば同じ三文字を入力すればいいと知っていた者の一人に、カリフォルニアのプログラマ、ドン・ホプ キンス（Don Hopkins）がいる。

「MIT が私やその他大勢に MIT のコンピュータを自由に使わせてくれたことを、私は永遠に感謝しますよ」とホプキンスは言う。「大勢の人間にとって、それは大きなことだったのです。」

ITS 時代には MIT の管理者側も大目に見ていたこのいわゆる「観光客」ポリシーは、10 研究所の ARPAnet へのプライマリ・リンクがオズになってからは失われた。当初はストールマンも仲間が後から同じ手順で入れるようにログイン ID とパスワードを同じにするポリシーを続けていた。しかし、オズの脆弱さのため、時間が経つにつれ、全くの偶然あるいは悪意をもってシステムをダウンさせる かもしれない部外者たちを閉め出すように管理者たちは動いた。この管理者たちから自分のパスワードを公開することを最終的に止めるよう求められると、ス トールマンは自分の倫理観を盾にしてそれを拒み、オズを使うことも一緒に止めてしまった。11

「MIT の AI ラボに初めてパスワードが登場した（ときに）、僕は自分の信念に従ってパスワードなしにする（ことを決めました）。」後に、ストールマンは語るだろう。 「なぜなら、コンピュータにセキュリティを施すのが本当に望ましいことだとは信じられませんでした。ぼくがセキュリティ体制を喜んで手助けする筋合いでは なかったのです。」12

ストールマンが巨大で強力なオズに従うのを拒絶したことは、1980 年代初期のハッカーと AI ラボの管理者との間に高まっていた緊張の現れだが、この緊張はハッカーのコミュニティ自身の内部で起きていた対立と比較すると色褪せてしまう。KL-10 の到来までに、ハッカーのコミュニティは既に二つの陣営に分裂していた。LMI 社とシンボリクス社である

シンボリクス社は、外部からの投資資金を使って多方面の AI ラボハッカーを募り、AI ラボの保護の外にある Lisp マシン・オペレーティングシステムのパーツを改良する仕事に配置した。1980 年の終わりまでに、Lisp マシンの自社ヴァージョンを開発するために 14 人の AI ラボ職員をパートタイム・コンサルタントとして雇った。残る少数は、ストールマンを別とすれば、LMI 社のために働いた。13 ストールマンは、プレッシャーのない AI ラボの生活を選び、どちらかの側に立つことを望まず、どちらの会社にも参加しないことに決めた。

最初は、他のハッカーたちは彼らの時間の一部を MIT で使い続け、MIT の Lisp マシン・オペレーティングシステムに貢献した。LMI 社とシンボリクス社はどちらも MIT からこれのコードをライセンスされていた。ライセンスは、両者が加えた変更について MIT に報告することを求めていたが、MIT にこれらの変更を再配布させることを求めていなかった。しかし、1981 年までは、それを許諾するという紳士協定が守られていて、彼らによる全てのシステムの改良は MIT ヴァージョンに含まれたので、Lisp マシンの全ユーザーが共有した。この状況は、MIT で両社がまだ中立的であるようにしていた。

ストールマンは 1982 年の 3 月 16 日という日付をよく覚えている。その日は彼の誕生日で、シンボリクス社の役員たちがこの紳士協定を終わらせることに決めた日だからだ。動機は LMI 社に対する攻撃だった。LMI 社が抱えるハッカーは少なく、全体の職員の数も少ないから、システムコードの改良を共有する主な利益は LMI 社が得ているのだとシンボリクス社の役員は考えた。彼らは、システムコードの共有を終わらせて、LMI 社を全滅させることを望んだ。そこで、シンボリクス社は、ライセンスに記された文字の力を行使することにした。LMI 社が利用できる、MIT ヴァージョンのシステムに彼らの改良を貢献するかわりに、MIT で走らせるユーザーのためには、シンボリクス社ヴァージョンのシステムを提供した。それを使った者は誰であれ、シンボリクス社にのみ検査のサービスを提供でき、もしも改良したときには、十中八九それらもシンボリクス社のためにのみ役立てられるだろう。

ラボの Lisp マシンを最新の状態に保つ責任のある（最初の二ヶ月間はグリーンブラットに手伝ってもらった）人間として、ストールマンは激怒した。シンボリクス社のハッ カーは数百の未完成コードを残していて、それはエラーを引き起こした。発表を「最後通牒」と見なした彼は、シンボリクス社から研究所へのマイクロ波通信を 切断して対抗した。それからシンボリクス社のマシンでは決して作業しないと宣言し、即刻 LMI 社への忠誠を誓った。「僕の見方では、AI ラボは第二次大戦のときのベルギーのような中立国です。」とストールマンは言う。「ドイツがベルギーに侵略してきた場合には、ベルギーはドイツに宣戦布告 して英仏に味方するんですよ。」

彼らの最新機能が依然として MIT の Lisp マシンのシステムに出現し、ひいては、LMI 社の Lisp マシンにもあることを知ったシンボリクス社の役員は気に入らなかった。ストールマンは著作権法の規定が何を求めているのか知っていたので、スクラッチから 機能を書き直した。彼は問題点の理解と改良のために、シンボリクス社が MIT に供給するソースコードを読む機会を利用し、そして、彼の加えた変更が全く別のやり方になることを確かめた。しかし、シンボリクス社の役員はこれが信じら れなかった。彼らはストールマンのコンピュータ端末に「スパイ用」プログラムをインストールして、ストールマンに不利な証拠を探そうとした。しかし、 MIT 当局を 1983 年初頭に彼らが訴えたとき、彼らは何の証拠も提出することができなかった。ソースの変更個所が 1 ダースあれば、両者のヴァージョンに同様に出現した。

AI ラボの管理者がシンボリクス社のずさんな状況証拠をストールマンに示したときに、彼はそれを否定して、類似性は実際にはフォークよりも前にあったものの残 存であることを示した。そして、彼はロジックを指摘した。もしも、彼の書いた数千行を持ち出した後で、これ以上、シンボリクス社がましな証拠を出せないな ら、コピーを避けるためのストールマンの入念な努力は有効だったとそれが証明したことになるのだ。AI ラボはストールマンの仕事を賞賛し、1983 年の終わりまで彼はそれを続けた。14 それでも、彼は実践の仕方を変更した。「完全に安全でいるために、もう向こうの[新機能や主な変更の]ソースコード

は読まなくなりました。ドキュメントだ けを使い、そこからコードを書いたのです。」最も大きな新機能のためには、シンボリクス社が文書をリリースするのを待っているよりも、それらを自分で設計 して、その後、シンボリクス社の文書が出たときに、その機能のためのシンボリクス社のインターフェイスとの互換性を追加した。それから、彼らが直したマイナーバグを見つけるためにシンボリクス社のソースコードの変更を読み、それらを違うやり方で直した。

経験はストールマンの決心を強化した。ストールマンはシンボリクス社の新機能を置き換えるための設計をしながら、AI ラボが MIT システムを使い続けるために、バグレポートが途切れないようにする AI ラボのメンバーからの援助も得ていた。MIT は、LMI 社が直接変更点にアクセスできるようにすることを継続していた。「何が何でも、シンボリクスを懲らしめてやるつもりでした。」とストールマンは語ってい る。そんな発言から分かることがある。それらはストールマンの非平和主義的な性格に光をあてるだけではなく、衝突が引き金をひいた強い感情のレベルも反映 している。

絶望が深かったのは、ストールマンがこれを「我が家」に対する「破壊」と見なしていたせいだ。つまり、AI ラボの結束の固いハッカーのサブカルチャーの終わりである。後にレヴィとの電子メールによるインタビューで、ストールマンは自分を歴史上の人物、1860 年代から 70 年代にかけてのインディアン戦争で絶滅させられた太平洋岸北西部のヤヒ族の最後の生き残り、イシになぞらえた。この喩えではストールマンの生き残りは叙事 詩やほとんど神話のような役回りだ。15 シンボリクス社のために働いているハッカーはそれを違う風に見ていた。シンボリクス社を皆殺し勢力と見るかわりに、多くのストールマンの同僚はそれを遅す ぎた妥当な試みとして見ていた。Lisp マシンの商品化の過程で、会社は、エンジニア主導のソフトウェア・デザインというハッカー方式を AI ラボという象牙の塔の境界を越えて企業の市場へと押し出した。そこでは経営者主導の方式が君臨していた。抵抗としてストールマンを見るより、多くのハッ カーは時代遅れの実践の代表として彼を見ていた。

個人的な敵意も状況に影響していた。シンボリクス社が AI ラボ職員のハッカーの大半を雇って連れ去っていく前ですら、後にシンボリクス社に参加するハッカーの多くは彼を避けるようになっていたとストールマンは言 う。「チャイナタウンに行くときに、もうぼくを呼ばなくなりました」とストールマンは回想する。「グリーンブラッドが始めた習慣では、誰かがディナーに出 かけるときは、ラボの誰かで他に行きたい者はいないかと聞いて回るか、メッセージを送るということになっていました。1980 年か 1981 年のあるときか ら、それが来なくなりました。彼らは僕を招かないだけじゃなくて、後に告白されたんですが、僕ぬきでディナーに行っているのを秘密にするように嘘をつく圧 力がかかっていたそうです。」

ストールマンは、この村八分のけちな方法に気持ちを傷つけられたものの、そこには何もなすべきことはなかった。シンボリクス社の最後通牒は、問題を 個人的な拒絶から広い範

囲に及ぶ不正に変えた。ライバルを倒す手段として、シンボリクス社がそのソースコードの変更の再配布をまったく許さないことにした とき、ストールマンは、シンボリクス社の目標を挫折させる決心をした。ストールマンは、MIT のオフィスにこもり、ソフトウェアのどの新しい機能にも改善にも同等のものを書くことにより、シンボリクス社のユーザーと同じ機能へのアクセスを、LMI 社のユーザーを含む、MIT システムのユーザーに与えた。

また、それはハッカー・コミュニティでのストールマンの地位を伝説的なものにした。すでに Emacs の仕事で名声を得ていたが、シンボリクス社のプログラマ全体のチーム - その中にはすでに伝説的なハッカー数名が含まれていた - の成果に匹敵するストールマンの能力は、情報時代における、更に言えば、どんな時代であれ、最大級の人間の才能である。それを「マスター・ハック」と呼び ストールマン自身を「コンピュータ・コードのヴァーチャル・ジョン・ヘンリー」と呼んだハッカーズの著者のスティーヴン・レヴィは、シンボリクスに雇われ たライバルの多くが、理想主義的なかつての同志に対して嫌々ながらも敬意を払うほかなかったことを注記している。結局、シンボリクス社で働くことにした ハッカーで、会社のパロ・アルトのオフィスにいたビル・ゴスパー（Bill Gosper）の、この時期のストールマンの成果に対する驚きの表明をレヴィは引用している。

ストールマンが書いたものを見て、それは駄目だと決めつけることができたかもしれません（おそらく違うが、誰かがそれは駄目だと信じ込ませることができたかもしれない）。そうだとしても、「しかし、ちょっと待った。あちらじゃ、ストール マンは一晩中誰とも議論していないんだ。一人で働いているんだぜ！これを一人でやれるなんてことは信じられない！」と言ったでしょう。16

ストールマンにとって、シンボリクス社に追いついて肩を並べる役割を演じていた日々は、誇りと深い悲しみの入り混じった感情を呼び起こした。第二次 世界大戦を戦った父親を持つ、生粋のリベラルの一人としてのストールマンは、平和主義者ではなかった。色々な意味で、シンボリクス戦争は十年前に AI ラボのスタッフに加わって以来のストールマンのキャリアに通過儀礼を提供していた。しかし、それはストールマンを十代の頃から養ってきた AI ラボのハッカー文化のトラウマ的な破壊と同時に起きた。ある日、コードを書くのを一休みしていたときに、ラボの機器室でトラウマ的な時間が過ぎて行くのを ストールマンは経験した。ストールマンはそこで図体の大きな使われなくなった PDP-10 マシンに遭遇した。眠っているようなライトにぎくっとした。かつては内部のプログラムの状態を表示するために黙ってコードを活動的に点滅していたライト だった。ストールマンは、愛する家族の一員の、良く保存されたミイラに出会ったのと感情的インパクトは変わらなかったと言う。

「機器室で泣いてしまいました。」とストールマンは言う。「そこにあるマシンが死んでいるのを見ていました。誰もフィクスする者はなくなって置き去りにされて、完全にコミュニティは破壊されてしまっているのを見ていたのです。」

ストールマンに悲しんでいる時間はほとんどなかった。それが巻き起こした熱狂と製作

に注ぎこまれた苦労の全てにもかかわらず、Lisp マシンは、テクノロジー市場における大規模な戦闘の前哨戦に過ぎなかった。コンピュータの小型化の無慈悲な速度は、より新しくより強力なマイクロプロセッ サを運んできた。まもなくそれは、現代的都市が古代の砂漠の村落を飲み込んで行くようにして、マシンのハードウェアとソフトウェアの能力を統合していくだ ろう。

何十万もの占有的ソフトウェア・プログラムがこのマイクロプロセッサの波に乗ったが、どれもユーザー・ライセンスのパッチワークと非開示契約 （NDA）で保護されていた。それはハッカーがソースコードをレヴューしたり共有したりするのを不可能にするものだった。ライセンスは粗雑で不釣り合いな ものだったが、1983 年までに、法廷を満足させるのに十分な程度に強力な、もぐり業者を追い出せるほどのものになった。かつては、高価なコンピュータ・システムという料理をよ りおいしくする付け合わせの一品だったが、ソフトウェアは、すぐにメイン・ディッシュになった。新しいゲームや機能に飢えたユーザーは、毎食後にレシピを レヴューする伝統的要求には注意を払わなかった。

パーソナル・コンピュータ・システムの領域ほど、この状況が明白なところはなかった。アップル・コンピュータ（Apple Computer）やコモドール（Commodore）は、組み込みのオペレーティング・システムのマシンを売ることによって、新たな百万長者を送り出し 続けていた。 ハッカー文化とハッカー文化がバイナリのみのソフトウェアを嫌うことは気付かれることがなく、パーソナル・コンピュータのユーザーの多くは会社がソース コード・ファイルの添付をやめることにしたときにそれに抗議する必要があるとはほとんど思わなかった。ハッカー倫理の無政府的信奉者でこの新しい市場にそ の倫理を持ち込もうとした者も少しはいた。しかし、たいていの場合、市場は新しいプログラムを書くのに十分な報酬を素早くプログラマに支払い、十分に抜け 目なく、ハッカー倫理をすっかり閉め出すエンド・ユーザー・ライセンス契約をそれらにつけた。

これらのプログラマの中で最も悪名高い人物の一人がビル・ゲイツ（Bill Gates）、ストールマンより二歳年下のハーヴァードのドロップアウトだ。その当時ストールマンは知らなかったが、ストールマンがメッセージを net.unix-wizards に送る七年前、ゲイツは、ソフトウェア開発者のコミュニティに自ら書いた公開状を送っていた。ゲイツは芽を出しかけた起業者で、アルバカーキに本社を置く ソフトウェア企業、マイクロソフト（Micro-Soft; のちに、Microsoft へ綴りを変更）の共同経営者だった。PC ユーザーのマイクロソフトのソフトウェア・プログラムのコピー行為に対する反応として書かれた、ゲイツの「ホビイストへの公開状」は、共有的なソフトウェ ア開発の観念を激しく非難した。

「見返りなしでプロフェッショナルな仕事をする余裕が誰にあるだろうか。」とゲイツは問いかける。「プログラミングに三人年かけ、すべてのバグを発見し、製品を文書化して、無料で配布できるのは、どんなホビイストだろうか。」17

AI ラボのハッカーは、ほとんど誰もその公開状を見ていなかったが、ゲイツの 1976 年の公開状は、商業的なソフトウェア企業と商業的なソフトウェア開発者の、双方のソフトウ

ェアに対する態度の変化を反映していた。市場が違うことを言って いるときに、どうしてソフトウェアをゼロ・コストの商品として扱うのか。1970 年代が 1980 年代に取って代わられるとき、ソフトウェアの販売は費用の回収以上のものになった。それは政治的声明になった。レーガン政権が、大不況に続く半世紀の間に 構築された連邦規制と支出プログラムの多くを廃止していったときに、少なくない数のプログラマがハッカー倫理を反競争的なもの、ひいては非アメリカ的なも のと見ていた。せいぜい、それは 1960 年代後半と 1970 年代初期の反企業的態度の復活だった。フレンチ・カフスのシャツとダブルのスーツの間に古い絞り染めのシャツが隠れていたのを見つけたウォールストリート の銀行家のように、多くのコンピュータ・プログラマは、ハッカー倫理を理想主義的な時代を思い出させる気恥ずかしいものとして扱っていた。

1960 年代全部を 1950 年代の復活に費やして過ごした人間として、ストールマンは同僚と一致しないことを気にかけなかった。しかし、ストールマンは、最良のマシンと最良のソフト ウェアで仕事をしてきたプログラマとして、彼が「率直な道徳的選択」と説明するほかなかったものに直面していた。すなわち、「占有的」ソフトウェアに反対 することをあきらめるのか、それとも、非占有的なソフトウェア・プログラムのシステムという別の選択肢の構築に自分の人生を捧げるのか、という選択であ る。占有的（proprietary ; プロプライエタリ）ソフトウェアというのは、ストールマンや彼の仲間のハッカーが、コピーや修正を制限する著作権の 条件やエンドユーザーライセンスがついた一切のプログラムを記述するために使っている用語である。シンボリクス社との厳しい試練の月日を経験した彼にとっ て、後者のオプションがより心地良く感じられた。「コンピュータを使うことをすっかりやめてしまうこともできたと思います。」とストールマンは言う。「ぼ くは特別な技能がありません。しかし、ウェイターになることはできたはずです。たぶん、しゃれたレストランではありませんが、どこかでウェイターになるこ とはできました。」

ウェイターをするというのは、つまり、プログラミングをすっかりドロップアウトしてしまうことで、それはコンピュータ・プログラミングの活動を完全 に諦めることを意味しただろう。それは彼に本当に多くの喜びを与えてきたものだった。ケンブリッジに引っ越して来て以来の人生を振り返り、ソフトウェア・ プログラミングが唯一の喜びを提供していた長い時期を容易に特定できることにストールマンは気付いた。ドロップアウトするよりも、ストールマンは最後まで 貫くことに決めたのだった。

無神論者のストールマンは、例えば、人生における宿命、ダルマ、天命といった観念を否定する。それにもかかわらず、占有的ソフトウェアを遠ざけて、 他人も同じことをするのを助けるオペレーティング・システムを構築する決心をするのは、自然な決心だと彼は感じている。結局、それはストールマン自身の、 頑固さ、先見性、コーディングの名人芸の個人的結合で、それは他の誰もが、存在するとは殆ど知らなかった道の分岐点に彼を思い至らせたのだ。1999 年の本、『オープンソース』のある章の中で彼の決心について記述して、ストールマンは、ユダヤの聖人ヒレルの言葉に要約された気持ちを引用している。

もしも、私が私自身のためにいるのでなければ、誰が私のためにいるのだろうか。もしも、私が私自身のためにのみいるのだというなら、私は何なのだろうか。もしも、今でなければ、いつだというか。12

聴衆に語りかけるとき、ストールマンは、宗教的なルートを避け、プラグマチックな言葉で決心について表現する。「自問してみました。一人のオペレー ティング・システムの開発者が、状況を改善するために何ができるだろうか。疑問を吟味してみると、問題を解決するのに必要とされているのは、まさに一人の オペレーティング・システム開発者だと悟るまで、それほど時間はかかりませんでした。」

ひとたび、その認識に達すると他の全てのことは「落ち着くところに落ち着いた。」とストールマンは言う。1983 年に、MIT はシンボリクスから第二世代の Lisp マシンを入手しようとしていた。その上では、おそらく MIT の Lisp マシン・システムは走らなかった。ひとたび、大半の MIT マシンが置き換えられてしまったら、ユーザーのバグ報告の不足で効率的にシステムを維持し続けることはできない。彼は止めねばならなかった。しかし、彼は 止めることを欲してもいた。MIT の Lisp マシン・システムは、自由ソフトウェアではなかった。ユーザーはソースコードを入手することさえ可能だったものの、それを自由に再配布することはできな かった。さしあたり、MIT システムの存続という目標はすでに達成されていた。LMI 社は生き残り、LMI 社自身のソフトウェアを開発し続けていた。

ストールマンは、彼の古いコミュニティを破壊した者たちを罰することに人生を全部使いたくなかった。彼は新しいコミュニティを建設したかった。彼は 彼の倫理的信条に妥協を求めるソフトウェアを公然と非難し、彼と他の人間がその拘束から脱することを容易にするプログラムを創り出すことに決めた。自由ソ フトウェア・オペレーティング・システムの構築を誓約しながら、「さもなくば、死か。もちろん、年とってからね」とストールマンは皮肉を言い、1984 年 1 月に GNU を構築するために MIT の職員を辞めた。

辞職は、ストールマンの仕事を MIT の法的庇護から遠ざけた。それでも、ストールマンは、MIT の施設を使い続けるとともに、後には、自分のオフィスを保有するのに十分な友人と同盟者を AI ラボの中に持っていた。また彼は、初期段階の GNU プロジェクトの費用を賄い、安心なものにするためのコンサルタント業務を外部でする能力も持っていた。しかし、MIT を辞職することによって、ストールマンは、利益相反関係や研究所のソフトウェア所有権に関するいかなる議論も無効化した。大人になる頃は社会的孤立の恐れ からAI ラボの抱擁の中へますますのめりこんでいった男は、今や、彼自身とその環境との間に法的なファイアウォールを築いたのだ。

最初の数ヶ月間、ストールマンは Unix のコミュニティからも同様に孤立して作業した。net.unix-wizards グループへの彼の告知は共感的反応を呼んでいたにもかかわらず、初期段階で十字軍への参加に応じる有志はほとんどいなかった。

「コミュニティの反応は、だいたい一致していました。」と当時の Unix ユーザー・グループのリーダーだったリッチ・モーリンは回想する。「『おお、それは素晴らしいアイデアだ

ね。コードを見せてくれよ。それができると示してくれよ。』とみんなで言っていたのです。」
　仕事は膨大であることが分かっていたストールマンは、可能であれば、どこにあるものであろうと、既存の自由ソフトウェアの再利用を試みることに決め た。そのため彼は GNU プログラムとツールに転換できる既存のプログラムとツールを探し始めた。最初の候補の一つが VUCK という名前のコンパイラだが、それは人気のあるプログラミング言語 C で書かれたプログラムをマシンで走行可能なコードに転換するものだった。オランダ語から翻訳すると、プログラムの頭字語は自由大学コンパイラ・キットを表 現していた。楽観的なストールマンは、プログラムの著者にプログラムは自由ですねと質問した。著者が、「自由大学」の言葉はアムステルダムにある Vrije Universiteit（自由大学）をさしているのだと告げたとき、ストールマンは悔しかった。
　「彼は馬鹿にした調子で『大学は自由です。しかし、コンパイラは自由じゃない。』と返事をよこしました。」とストールマンは回想する。彼は助けるの を断ったばかりか、GNU を開発する計画から手を引いて、そのかわりに VUCK の販売の強化になるアドオンでも書いたらどうかとストールマンに提案した。見返りに利益を分けるという。「だからこそ、GNU プロジェクトのためのぼくの最初のプログラムは、マルチ言語、マルチプラットフォームのコンパイラにしようと決めました。」19
　VUCK のかわりに、ストールマンは、ローレンス・リバモア国立研究所（Lawrence Livermore National Lab）のプログラマが書いた、パステル（いかがわしいパスカル）・コンパイラ（the Pastel compiler）を見つけた。彼らの言によれば、ストールマンにプログラムを贈ったときにコンパイラはコピーや修正が自由だった。不運にも、プログラム はジョブに適さなかった。なぜなら、そのメモリ要件は巨大だったのだ。それは入力された全ファイルをコア・メモリーで構文解析した。そしてコンパイルし終 えるまで全ての内部データを保持し続けた。メインフレーム・システムでは、この設計は許容された。Unix システムでは機能を喪失させる障害だった。Unix を走らせる 32 ビットのマシンでさえ、しばしば、プログラムにそれだけのメモリーを提供できなかったからだ。ストールマンは、当初、コンパイラに C 互換のフロントエンドを構築して、それを大きなメモリー空間を扱えるシステムである、より大きな Vax でテストして、かなり作業を進めていた。彼は、68010 にシステムを移植しようとしたときに、なぜそれがクラッシュしたかを調べて、メモリーの大きさの問題を発見した。そして、すっかり新しいコンパイラをスク ラッチから構築する必要があるという結論に達した。ストールマンは、結局、これをやった。GNU C コンパイラ、すなわち、GCC である。しかし、1984 年には、コンパイラについて何をすべきか明らかではなかった。そこで、GNU の他の部分に集中している間に、それらのプランが固まっていくのにまかせようと決めた。
　かくして、ストールマンは、1984 年の秋に Emacs の GNU ヴァージョンの開発を始めた。彼自身が十年にわたって監修してきたプログラムだ。Unix コミュニティに、二つあるネイティブのエディタ・プログラムは、サン・マイクロシステムズの共同設立者のビル・

ジョイ（Bill Joy）の書いた vi と、ベル研究所の科学者（そして Unix の共同創作者）のケン・トンプソン（Ken Thompson）の書いた ed だった。どちらも便利で人気があったが、どちらも、Emacs の際限なく拡張可能な性質は提供していなかった。

当時を振り返ってストールマンは、戦略的条件の観点から見ていたのではなかったと言う。「Emacs が欲しかったんです。開発をする良い機会でした。」

ここでもまた、ストールマンは時間を節約することを望んで既存のコードを見つけていた。Emacs の Unix ヴァージョンを書くにあたって、まもなく、ストールマンは、ゴスリング Emacs とか GOSMACS と名付けられた、Emacs の C をベースにしたヴァージョンの作者であるカーネギー・メロンの大学院生、ジェームス・ゴスリング（James Gosling）の踏み跡をたどって行った。Emacs のゴスリングのヴァージョンは、MOCKLISP と呼ばれる簡略化した lisp 言語の派生物を利用したインタープリタを含んでいた。ゴスリングは、GOSMACS を著作権の下に置き、個人的に所有しているソフトウェア会社、ユニプレス社にその権利を売却していたが、ストールマンは、初期の MOCKLISP インタープリタに参加していた仲間の開発者の保証を援用していた。その開発者によれば、ゴスリングはカーネギー・メロンの博士課程の学生だったときに、 コードへの貢献と引き換えに、協力者自身の GOSMACS のヴァージョンの配布の許可を電子メールで与えていた。

当初、ストールマンは、オリジナルの PDP-10 Emacs との完全な互換性を実装するために、ユーザーレベルのコマンドだけを変更しようと思っていた。しかし、実際の Lisp と比較すると Mocklisp がいかに非力かが分かって、彼はそれを本物の Lisp システムと置き換えざるをえないと感じた。これは当然ながら、Gosmacs のハイレベルコードの大半を、Lisp のより強力な力と柔軟なデータ構造を生かして、全く違ったやり方で書き直させるものだった。1985 年の中頃までに、インターネットでリリースされた GNU Emacs の中には、ほんの幾つかのファイルに Gosmacs からの残りコードがあるだけだった。

ユニプレス社はストールマンのプロジェクトに気付き、他の開発者が独自ヴァージョンの Gosmacs を配布する許可を受けていることを否認した。その開発者は自分の主張を裏付ける古い電子メールのコピーを見つけることができなかった。ストールマンは、こ の問題を Gosmacs から残っていた僅かなモジュールを置き換えるものを書いて解消した。

しかし、それでも、開発者によるソフトウェアの権利の売却という観念や、そもそも開発者にソフトウェアを売る権利があるという観念に、ストールマン は腹が立った。1986 年のスウェーデン王立工科大学での講演で、ストールマンは、占有的ソフトウェアに伴う危険性のもう一つの例としてユニプレス社の事件を引用している。

「時々、思うことがあるんです。ひょっとして、自分の人生でできた最善のことは企業秘密になっている占有的ソフトウェアを大量に見つけて、そのコ ピーの街頭配布を開始することなんじゃないだろうか。そうすれば、企業秘密なんかじゃなくなってしまう。」とストールマンは言った。「ひょっとして、実際 に自分で書いた新しい自由ソフトウェアを人にあげるよりも、自分にとってもっと効率的なんじゃないだろうか。しかし、怖いからそれは誰

も受け取ることさえ できません。」20

そのストレスにもかかわらず、ゴスリングのコードに関する論争は、長期的には、ストールマンと自由ソフトウェア運動の両方を助けることになる。それ は問題のある派生物の出現を可能にしていた Emacs コミューンの弱点とインフォーマルな信頼のシステムの弱点にストールマンが取り組むことを強いた。それはまた、自由ソフトウェア運動の政治的目標を明確に することをストールマンに強いた。1985 年の GNU Emacs のリリースに続いて、ストールマンは、1983 年 9 月に投稿したオリジナルの声明を拡大した「GNU 宣言」を発表した。ストールマンは、占有的ソフトウェア・プログラムの横行を正当化するために商業的プログラマやアカデミックなプログラマが使っている多 くの議論を扱った長い章をその文書の中に含めた。その中の「プログラマは創造性で報酬に値するのではない」という論考は、少し前のゴスリン Emacs のエ ピソードに関するストールマンの怒りを要約した回答を提供している。

「もしも報酬を受けるに値するものが何かあるとすれば、それは社会的な貢献である。」ストールマンは書いた。「創造性は社会的貢献でありうる。しか し、それは社会がその結果を自由に使える限りにおいてのことだ。もしもプログラマが技術革新的プログラムの創造により報酬を受けるに値するとすれば、同じ 理由により、もしもこれらのプログラムの使用を制限するならば、罰せられるに値するのである。」21

GNU Emacs のリリースとともに、GNU プロジェクトはついに見せることのできるコードを持った。それはまた、いかなるソフトウェア・ベースの会社にとっても心配の種だった。Unix 開発者がそのソフトウェアを楽しみ始めるにつれて、お金、贈り物、そしてテープの注文が殺到し始めた。GNU プロジェクトのビジネスの面に取り組むべく、ストールマンは仲間から何人かを選抜して、自由ソフトウェア財団(FSF; Free Software Foundation)、GNU プロジェクトをその目標に向かって加速するための非営利組織をたちあげた。ストールマンを理事長、変化に富むハッカーの同盟者を理事として、FSF は、GNU プロジェクトの法人的側面を提供する役目をした。

ロバート・シャッセル(Robert Chassell)は、Lisp マシン社で働いていたときに、ストールマンとディナーで話をした後、自由ソフトウェア財団の五人の設立委員の一人になったプログラマだ。シャッセルは、財 団の会計も担当している。その役割は始めたときは小さかったが、すぐに大きくなった。

「85 年の支出と収入は、全体でおよそ 23,000 ドル程度だったと思います。」とシャッセルは回想する。「リチャードには自分のオフィスがあって、我々は場所を借りていました。私は何でもかんでも、とく にテープは、私のデスクの下に置いていました。しばらくして、テープやその手のものを保管できる場所を LMI 社が貸してくれたんですが、それでもそうしていましたね。」

自由ソフトウェア財団は、面の提供に加えて、中心も、幻想から醒めた他のプログラマたちのための重力の中心も提供した。非常に平等な関係に見えてい た Unix 市場は、ストー

ルマンが最初の GNU の発表をしたときでさえ、ますます競争的になっていた。顧客の繋ぎ止めを強化しようとして、会社は、Unix のソースコードへのアクセスを閉ざし始めていた。進行中の GNU ソフトウェア・プロジェクトへの照会をただ加速するばかりの形勢である。かつてはストールマンをやかましい変人だと思っていた Unix のウィザードたちは、自分が突き止めた問題から自由になることへの希望または絶望に応じて、今や、ストールマンをソフトウェアの預言者またはソフトウェア のカッサンドラとして見始めていた。

「多くの人は、自分の身に起きるまで気が付きません。奪われてしまうだけのソフトウェアの仕事に何年も費やしたら、どんなにがっかりさせられること でしょうか。」FSF を設立して間もない頃の寄稿者達の感情と意見を要約してシャッセルは言う。「そんなことが二度も起きると、自問し始めるんです。『おい、ちょっと待て よ。』ってね。」

シャッセルにとって、自由ソフトウェア財団に参加することを決めたのは、彼自身が経験した個人的喪失感からだった。LMI 社の前に、シャッセルはケンブリッジ地区のソフトウェア会社、カドマス社に雇われて Unix の入門書を書いていた。カドマス社が破産したときにその本に対する権利も失われてしまった。シャッセルは権利を買い戻そうとしたが成功しなかったと言う。

「私の知る限り、あの本はまだどこかの棚の上にあって、使用不可、コピー不可で、ただシステムから除外されただけになっているのです。」とシャッセ ルは言う。「自分で言うのも何ですが、実に良い入門書でした。三ヶ月か四ヶ月かければ、今日の GNU/Linux の入門として完全に使えるものに変換できるかもしれません。私の記憶に残っているものを別にすれば、その全経験が失われてしまったのです。」

かつての雇い主が破産と闘っている間に自分の著作が泥沼へ沈んで行くのを見ざるを得なかったシャッセルは、ストールマンを脳卒中にしそうな怒りがわ かる気がしたと言う。「私にとって、一番はっきりしているのは、ちゃんとした生活をしたい人なら自分の人生の一部から閉め出されるのを望むはずがないとい う感覚です。」シャッセルは言う。「何かを始めて、それを完成し、そしてまた修正する自由を持っている、その何かが何であれ、この、そういう理念の全体 が、本当の違いをつくりだすのです。数年かけて価値のある仕事を達成した人生の後で、幸せだったとそれは考えさせるでしょう。なぜなら、さもなければ、奪われ、放棄され、または控えめに言っても、それに関わることができなくなってしまうのです。人生の一部が失われてしまうようなものです。」

後注

1『最初の GNU 声明』(1983 年 9 月) リチャード・ストールマン、参照。

2 同書。

3 同書。

4『バークレー Unix の 20 年』マーシャル・カーク・マクージック、「オープンソース」(株式会社オライリー・ジャパン、1999 年) を参照。

5 リチャード・ストールマン（1986 年）参照。

6 同書。

7 複数の出典を参照。リチャード・ストールマンのインタビュー、ジェラルド・サスマンの電子メール、ジャーゴン・ファイル 3.0.0

http://catb.org/jargon/html/T/TWENEX.html

8 http://www.as.cmu.edu/ geek/humor/See_Figure_1.txt.

9 リチャード・ストールマン（1986 年）参照。

10『MIT AI ラボの観光客ポリシー』を参照。

http://www.art.net/ hopkins/Don/text/ ~ tourist-policy.html

11 リチャード・ストールマン（1986 年）参照。

12 同書。

13 Steven Levy, Hackers (Penguin USA [paperback], 1984): 423. 『ハッカーズ』スティーブン・レビー（工学社）

14 H. P. ニューキストの『頭脳制作者たち』が、AI ラボはストールマンに Lisp マシンプロジェクトから離れているように告げた、と言うのは誤りである。

15『ハッカーズ』で、スティーブン・レヴィがこの時期を念頭に記述したときに、ストールマンを「最後の真のハッカー」としていたが、彼の意図した 意味はあなたが思ったものと違ったかもしれない。レヴィは「真のハッカー」という用語を MIT のハッカー・コミュニティを他の二つのハッカー・コミュニティから区別するために使い、後者は本の中で後で解説して、別の名前を与えた。[「真のハッ カー」は第一部のタイトルで MIT ハッカーを指している。第二部はハードウェア・ハッカー、第三部はゲームハッカー。（訳者）]ストールマンだけを残し て、このコミュニティが溶解したとき、彼は、結果として、最後の「真のハッカー」になった。レヴィは、他の者は誰も本当のハッカーではないと言おうとして いるのではない。しかし、人々は、特にレヴィの本の説明を読んでいない人々は、彼の言葉をそのように解釈する傾向がある。ストールマン自身は、レヴィの本 のそれらの言葉を使って自らを形容したことはない。

16 Steven Levy, Hackers (Penguin USA [paperback], 1984): 426 日本語訳は、『ハッカーズ』スティーブン・レビー（工学社）

17『ホビイストへの公開状』ビル・ゲイツ（1976 年 2 月 3 日）。この公開状のオンライン上のコピーは次のサイトで見ることができる。

http://en.wikipedia.org/wiki/Open_Letter_ to_Hobbyists.

18 次を参照。ストールマンはこの声明に脚注を加えて、「無神論者の一人として、宗教指導者に従うことはないが、彼らの中の或る者が述べた言葉に感心している自分に気付くことがある。」と書いている。

http://www.gnu.org/gnu/the-gnu-project.html.

19 Richard Stallman, "The GNU Operating System and the Free Software Movement,"

Open Sources (O'Reilly & Associates, Inc., 1999): 65.を参照。日本語訳は、「GNU システムとフリーソフトウェア運動」、リチャード・ストールマン、『オープンソース』（株式会社オライリー・ジャパン、1999 年）。

20 リチャード・ストールマン（1986 年）

21『GNU 宣言』（1985 年）リチャード・ストールマン

http://www.gnu.org/gnu/ manifesto.html

第八章　聖イグヌーティウス

マウイ高性能計算センター(MHPCC)は、キヘイの町のすぐ上の赤茶けた丘の一階建てのビルにある。100 万ドルの景色と不動産価格数 100 万ドルのシルバーワード・ゴルフコースに囲まれたこのセンターは科学的浪費の極みに見える。テック・スクウェアのような四角くて不毛な領域、ましてやイリ ノイのアルゴンヌやニューメキシコのロスアラモスのような拡大する研究都市とはほど遠く、MHPCC は、科学者たちが、博士号取得後の研究プロジェクトよりも、日焼けにより多くの時間を費す場所であるかのように見える。

このイメージは真実の半分でしかない。MHPCC の研究者たちは確かに、この地方のリクリエーションの機会を利用しているが自分たちの仕事にも真剣に取り組んでいる。世界で一番強力なスーパーコンピュー タを追跡しているウェブサイトの Top500.org によれば、MHPCC に設置されている IBM SP Power3 スーパーコンピュータは、毎秒 8,370 億回の浮動小数点演算のペースで動作し、世界に 25 台あるもっとも強力なコンピュータの一つになっている。ハワイ大学とアメリカ空軍との共同所有のもとで運用されているこのマシンは、軍の兵站と高温物理学 の研究に伴う膨大な数値計算タスクの間で、計算サイクルを分けあっている。

簡単に言えば、MHPCC は、科学や工学という頭脳文化とハワイ諸島のくつろいだ文化が、平和にバランス良く共存しているユニークな場所だ。研究所のウェブサイトの 2000 年のスローガンは、これを「天国でのコンピューティング」と要約している。

ここは確かにリチャード・ストールマンを見出しそうな場所ではない。この男は、職員オフィスの展望用の窓越しに近くの美しいマウイ海峡の風景を眺め て、「日が当たり過ぎだ」などと、そっけない批評をつぶやく。それでも、あるコンピューティングの天国から、もうひとつのそれへの使者として、たとえ、自 分のハッカーの瞳を厳しい太陽の眩しい光に晒すことになろうとも、ストールマンには伝えるべきメッセージがある。

私がストールマンの講演に間に合うように着いたときは、会議室は既に一杯になっていた。性別の構成はニューヨークの講演よりは幾らかましで、男性 85％、女性 15％ だったが、それほどましなわけでもない。聴衆の半分は、カーキ色のパンツにロゴ入りのゴルフシャツを着ている。他の半分は、地元のスタイルだ。世界のこの 一角では一般的な、派手な花柄のシャツに身を包み、顔は濃い黄土色の色調を帯びている。唯一、おたくっぽさの名残りを感じさせるのは、ノキアの携帯電話、 パーム・パイロットそしてソニーの VAIO ラップトップといった機器だけだ。

言うまでもなく、青の無地の T シャツ、茶のポリエステルのスラックス、白のソックスという出立ちで、部屋の前に立つストールマンは、ひどく目立っている。会議室の蛍光灯が、彼の陽に当 たっていない皮膚の不健康な色を際だたせるのに一役かっている。1 彼のひげと髪では、ハワイの一番涼しい地域でも、汗がしたたってくるだろう。額に「本土人」といういれずみしているわけではないが、そうしたとしても、今 より以上によそ者じみて見え

ることはあるまい。［RMS: 他と違って見えるのは何か悪いことがあるのだろうか。］

ストールマンが部屋の前でぶらぶらしているとき、マウイ FreeBSD ユーザ・グループ（MFUG）のロゴの入った T シャツを着た二、三の聴衆が、カメラと音声機材の設定を急いでいた。FreeBSD は、1970 年代の由緒あるアカデミック版 Unix であるバークレー・ソフトウェア・ディストリビューションから派生した自由ソフトウェアで、技術的には、GNU/Linux オペレーティング・システムと競合している。それでも、ハッカー界では、ストールマンの講演は、グレートフル・デッドとその伝説的に有名な、ファンによる 録音アーカイブの熱気の余韻を偲ばせながら、記録されている。ハンブルグやムンバイやノボシビルスクにいる仲間のプログラマたちが、ストールマンの知恵の 最新の精華を見逃さないようにしてあげることは、この地域の自由ソフトウェアの代表者である MFUG のメンバーの肩にかかっている。

グレートフル・デッドという比喩はぴったりである。自由なソフトウェア・モデルに本来備わっているビジネスチャンスを説明するのに、ストールマンは よくグレートフル・デッドを取り上げてきた。ライブ・コンサートをファンが録音できないように規制することを拒否したために、グレートフル・デッドはただ のロック・グループ以上のものになった。彼らは、グレートフル・デッドの音楽に捧げられた同族的コミュニティの中心になった。時とともに、この同族的コ ミュニティは非常に大きくなり、また献身的だったので、このバンドはレコード契約を避けて、音楽ツアーとライブ出演だけでやって行くことにした。1994 年、バンドのツアー活動最後の年には、グレートフル・デッドは入場料だけで、5,200 万ドルを引き寄せた。2

ソフトウェア企業でこれほどの成功を納めることができたものはほとんどなかったが、自由ソフトウェア・コミュニティの同族的側面は、1990 年代後半に多数の人々が、ソフトウェアのソースコードを公開することは良いことかもしれない、という考えを受け入れ始めた理由の一つだ。自分に忠実な従者 たちを作り出そうとして、IBM、サン・マイクロシステムズ、ヒューレット・パッカードといった企業は、ストールマンの精神ではないにしても、その字句を 許容するようになった。GPL を情報技術産業の「マグナ・カルタ」と形容することで、ZDNet のソフトウェア・コラムニスト エヴァン・ライボヴィッチ（Evan Leibovitch）は、全てを GNU にすることへの愛好心の高まりを、一時的な流行以上のものと見ている。「マグナ・カルタが英国臣民に権利を与えたように、GPL は、コンピュータ・ソフトウェアのユーザーのために、消費者の権利と自由の力を与えるのだ。」3

自由ソフトウェアのコミュニティがもつ同族的な側面からも、そうでなければ物理プロジェクトで働いたり、ウィンドサーフィンの気象情報を見にウェブ サーフィンをしていたかも知れない 40 数人ものプログラマが、なぜストールマンの講演を聞きに会議室に詰めかけたのかが説明できる。

ニューヨークの講演とは異なり、ストールマンは導入部を置かない。自己紹介もしない。FreeBSD の人たちが装置を設定し終り、動かし始めると、ストールマンは前に進みでて話

し始め、部屋中の他の全ての声を平らにならす。
「ソフトウェアの利用について、社会はどのようなルールを持つべきか、という問題を考えるときに、人々はほとんどの場合、ソフトウェア会社の立場か ら問題を考えています。そして、自己中心的な観点からその問題を考えるのです。」と言って、ストールマンは講演を始める。「どんなルールを他の人々に課し たら、我々に多くのお金を支払わなければならないようにできるのだろうか？ ぼくは好運にも、1970 年代に、ソフトウェアを共有するプログラマのコミュニティに属していました。このため、この同じ問題を常に別の角度から見ようとします。どんなルールな ら、そこに暮らす人々にとって良いような、良い社会を可能にするのだろうか？ そして、全く別の答えにたどりついたのです。」
またもや、ストールマンは間をおかずにゼロックスのレーザー・プリンタの話に入っていき、少しの間、聴衆に例の芝居がかった指さしのジェスチャーを振りまく。また、一、二分をグヌー・リナックスという名前に費す。
「僕にこう言う人がいます、『このシステムへの功績を認めさせるのに、何でそんなに騒ぎ立てるんです？ つまるところ、重要なのは仕事をかたづけることであって、あなたの功績が認められるかどうかじゃないでしょう。』うん、それが本当だとしたら、これは賢明 なアドバイスだったでしょう。でも、仕事はオペレーティング・システムを構築することではなかったんです。仕事は、コンピュータのユーザーの自由を広げる ことなのです。そして、そうするには、コンピュータで自由に何でもできるようにしないといけない。」[4]
ストールマンはつけ加える、「やることは、まだ山ほどあります。」
聴衆の一部にとって、これは聞き古した話題だ。他の一部にとって、それはいささか難解だ。ゴルフシャツの一団の一人が寝息を立て始めたとき、ストールマンは話すのを止め、誰かがその人を起こしてくれるように頼んだ。
「以前ある人が、ぼくの声には心を和ませるところがある、と言い、ヒーラーか何かなのか、と尋ねられたことがあります。」とストールマンは言い、聴 衆からすばやく笑いを引き出す。「思うに、それは多分、至福の、リラックスした眠りにただよいつくのを助けてあげられる、ということだと思います。で、君 たちの中にもそれを必要としている人がいるんでしょう。ぼくは、そうするのをじゃまするべきじゃないと思います。もしも眠りが必要だったら、何を差しおい ても眠るべきです。」
講演は、ソフトウェア産業内部でも、自由ソフトウェア・コミュニティ内部でも、次第に関心の高まっている問題であるソフトウエア特許について簡単に 論じて終る。ナップスターと同様に、ソフトウェア特許は、物理的世界のために書かれた法律や概念を、情報技術という摩擦力の働かない宇宙に適用しようとす るときの厄介な性格を反映している。
著作権法と特許法は違う働きをする。そしてソフトウェアの領域に全く違う効果を及ぼす。プログラムの著作権は、そのプログラムのコードの複製と改変 をコントロールする。そしてプログラムの著作権は、プログラムの開発者に帰属する。しかし、著作権はアイデアをカバーしない。言い換えれば、開発者は、著 作権の下では、既存のプログラムに見られ

る機能やコマンドを自分自身のコードで実装する自由がある。それらの局面はアイデアであって、表現ではなく、従っ て、それらは著作権法の範囲外だからである。

同様に、バイナリ・プログラムの働きを解読し、違うコードで同じアイデアとアルゴリズムを実装するのは、大変な仕事ではあるが、合法だ。この仕事は「リバース・エンジニアリング」として知られている。

ソフトウェア特許の働きはそれとは異なる。アメリカ特許庁によれば、会社や個人は、コンピュータの使用に関する新規なアイデア（すなわち、少なくと も特許庁に知られていないアイデア）について特許権を取得できる。理論的には、このことにより、特許権者は、特許出願後最低 20 年間、その発明の開示とひきかえに、一定の独占権を認められる。実際には、プログラムの動作はたいてい自明なので、そして、いずれにせよリバース・エンジ ニアリングにより決定できるので、この開示は限定的な価値しかない。著作権と異なり、特許により、その保有者は、同一または類似の機能をもつソフトウェ ア・プログラムを他人が独自に開発することを阻止できるようになる。

市場の全ライフサイクルが 20 年でカバーできてしまうソフトウェア産業では、特許は戦略的な重みをもつ。かつて、マイクロソフトやアップルのような企業が、著作権と各種技術の「ルック アンドフィール」をめぐって争っていたところで、今日のインターネット企業は、個々のアプリケーションやビジネス・モデルに関する利益を確保する一手段と して特許を使う。もっとも有名な例は、2000 年にアマゾン・ドット・コムが、同社の「ワンクリック」オンラインショッピングの手順を特許化しようと企てたことだ。だが、大部分の企業にとって、ソフト ウェア特許は防御手段になっている。企業特許の一セットを、もう一方のそれとバランスさせるような相互利用許諾の取引が、企業間の緊張緩和のために、恭しくとり行われる。とはいえ、コンピュータによる暗号化や画像アルゴリズムといった二、三の名高いケースでは、ソフトウェア・ベンダーは、ライバルの技術を 妨害することに成功している。たとえば、アップル社からの特許の脅しにより、自由ソフトウェアから幾つかのフォント描画機能が失われている。

ストールマンにとって、ソフトウェア特許の問題は、ハッカーが絶えず警戒心をもつことの必要性を如実に示すものだ。それは、自由ソフトウェア・プロ グラムの競争的利点にも優る政治的利点を強調することの重要性に注意を促すことでもある。ストールマンは、競争可能な性能と価格という二つの領域は、 GNU/Linux や FreeBSD のような自由なオペレーティング・システムがその占有的対応物に対して既にもっている際だった長所だが、ユーザーと開発者の自由という大問題に比べたら、 それは副次的なものにすぎないと言う。

この立場は、コミュニティ内の論点になっている。オープンソースの主唱者は、自由なソフトウェアの政治的長所より、利用上の長所を強調する。自由ソ フトウェアの政治的意義を強調するかわりに、オープンソースの主唱者たちは、ハッカーの開発モデルの工学的完全性を強調することを選んだ。ピアレビューの 強力さを引き合いに出して、オープンソースの議論は、GNU/Linux や FreeBSD のようなプログラムを、良く作られ、良く検査され、

したがって、平均的なユーザーにとって、より信頼できるものと特色づける。
　このことは、「オープンソース」という用語が、政治的な意味合いを持たないということではない。オープンソースの主唱者にとって、オープンソースと いう用語は、二つの目的に仕える。第一に、それは「フリー」という言葉にまつわる混乱を取り除く。この言葉は多くのビジネスで、「費用ゼロ」を意味すると 解釈される。第二に、それは企業が自由ソフトウェアという現象を、倫理ではなく、技術に基づいて検討できるようにする。オープン・ソース・イニシアチブの 共同創設者であり、この用語を支持する指導的ハッカーの一人であるエリック・レイモンドは、『黙ってソースコードを見せな』と題した 1999 年のエッセーで、ストールマンに従って政治的な道をたどることのフラストレーションについて要領よくまとめている。
　ストールマンのレトリックは、我々のような人間には、とても魅惑的だ。我々ハッカー は、思考家であり理想主義者であり、「原則」だの「自由」だの「権利」だのに訴えかけるものに共鳴しやすい。彼の計画の詳細には同意できなくても、我々は ストールマンのレトリックスタイルが有効であってほしいと思い、我々はそれが有効なはずだと考え、それが我々ほどはまりこんでいない 95 %の人々に対して失敗すると、戸惑い、信じられないと思う。5
　この 95 %には、とレイモンドは書く、企業の経営者、投資家、ハッカー以外のコンピュータ・ユーザーの一群がいて、ひたすら数の重みで商業的なソフトウェア市場の 全体的な動向を左右する傾向がある。こういう人々を獲得する道がないと、とレイモンドは論じる。プログラマは社会の片隅で、そのイデオロギーを追求するよ うに運命づけられたものでしかなくなる。
　ストールマンが「コンピュータ・ユーザーの権利」を語ることに固執するとき、彼は、我々が古い間違いを繰り返すように、危険で魅力的な手招きをしているのだ。この手招きを我々は拒否すべきだ -- それは、彼の原則が間違っているからではなく、その種の言語をソフトウェアに適用しても、我々以外の誰も説得できないからだ。事実、それは我々の文化の外 側の大部分の人を混乱させ、追い払う。6
　しかしながら、レイモンドの前提をストールマンは退ける。
　我々が失敗していないのに我々の失敗を説明するというレイモンドの試みは、ミスリーディングだ。我々の目標は大きく、道は長いけれど、すでに歩いてきた道も長い。ハッカーではない人の価値観に対するレイモンドの悲観的な断定は誇張してい る。ハッカーではない多くの人が、オープンソースが重視した技術的利点以上に、我々が焦点を合わせた政治的問題に関心を持っている。全ての国々ではない が、これには、しばしば政治のリーダーも含まれている。エクアドルとブラジルの大統領を促して、自由ソフトウェアを政府機関に導入させたのは、「もっと良 いソフトウェア」ではなく、自由ソフトウェアの倫理的な理想だった。彼らはギークではないが、自由を理解するのである。
　しかし、ストールマンによれば、オープンソースの論法の一番の欠陥は、それが弱い結論に導くことだ。それは自由ソフトウェアを実行するように多くの ユーザーを説得する。し

かし、自由ソフトウェアに完全に移行する根拠を提供しない。これはユーザーに部分的に自由を与えるが、自由を認識することを教え ず、自由それ自体に価値を認めない。だから、ユーザーは依然として自由をその手から落として失ってしまいそうなままになっている。たとえば、自由ソフト ウェアの改良が特許によってブロックされたときに何が起こるだろうか。

多くのオープン・ソースの提唱者は、ソフトウェア特許に反対することにかけては、ストールマン以上とはいかなくても、同程度には声高である。多くの 占有的ソフトウェア開発者にとってもそうであるのと同様に、特許は、彼らのプロジェクトも脅かしているからだ。しかし、ソフトウェア特許がソフトウェアの 機能の領域を立ち入り禁止にしがちであることを指摘しながら、ストールマンは、そのようなケースでの自由ソフトウェアの理想とオープンソースの含意とを対 比する。

「これは、ぼくらがもっと良いソフトウェアを作る才能が無いからじゃないんです、」とストールマンは言う。「ぼくらにそうする権利が無いということ なんです。ぼくらが公共にサービスするのを禁じている誰かがいるのです。そこで、ユーザーが自由なソフトウェアのこういうギャップに遭遇すると、何が起こ るでしょう？ たとえば、ユーザーが、オープンソースの運動によって、こうした自由の良い点は、それがもっと強力で信頼できるソフトウェアにつながるからだ、ということ を説得されていたとしたら、ユーザはこう言うでしょう、『あなたは、約束したものをくれないじゃないか。このソフトウェアは、もっと強力じゃない。それに は、これこれの機能がない。あなたは嘘をついていた。』だが、ユーザーが、自由ソフトウェアの運動に同意していたなら、自由はそれ自体に意義があるのだか ら、彼らは言うでしょう、『なんで、こうした機能を手にする私の自由を妨げるんだろう。』そして、こういう反響とともに、ぼくらは、こうした特許の爆発か ら受ける打撃を生き延びることができるかもしれません。」

ストールマン本人がその政治的メッセージを発信するのを観察しても、混乱させるものや不愉快なものを見つけるのは困難だ。ストールマンは、見掛けは 不快かもしれないが、メッセージは論理的だ。聴衆の誰かが、占有的ソフトウェアを避けていると、自由ソフトウェアの支持者達は、最新の技術的進歩に追いつ いていく能力を失うのではないか、と尋ねると、ストールマンは彼の個人的な信条によって、その質問に答える。「自由は、単なる技術的進歩より重要だと僕は 思う、」と彼は言う。「進歩しているが自由でないプログラムより、あまり進歩していなくても、自由なプログラムを僕はいつでも選ぶでしょう。なぜなら、ぼ くは、そのために自由を放棄したくないからです。ぼくのルールはこうだ。それを君と共有できないなら、ぼくはそれを入手しない。」

倫理を心に帯びた人々といえば信仰を意味するので、こういう回答は、ストールマンのメッセージがもつ宗教風な性格を強める。しかしながら、ユダヤ教 徒が律法に適った食事を守り、モルモン教徒が飲酒を拒むのと異なり、ストールマンは戒律に従うのではなく、単に自由を譲り渡すことを断る。彼の講演は、そ うするための実践的要件を説明する。すなわ

ち、占有的プログラムはあなたの自由を奪い去る。だから、もしも自由が欲しければ、そのプログラムを拒絶する必 要がある。

ストールマンは、占有的ソフトウェアの代りに自由ソフトウェアを使うという自分の決心を、他人も共有して欲しいと彼が望んでいる個人的信条の色で染 める。ソフトウェアの伝導者として行動するとき、ストールマンは、こうした信条を聞き手に押しつけることはない。とはいえ、聞き手がソフトウェアの道理に 至る真の道がどこにあるかを知らないまま、ストールマンの講演を後にすることも滅多にない。

このメッセージを強く心に刻みつけるためか、ストールマンは、講演を一風変わった儀式で締めくくる。プラスチック製の食料品バッグから黒いローブを 取り出し、ストールマンは、それをはおる。もう一つのバッグから、光を反射する黄色いコンピュータディスクを取り出し、それを頭に載せる。聴衆はびっくり して笑い出す。

「我は Emacs 教会の聖イグヌーティウスなり、」右手を祝福するときのように上げて、ストールマンは言う。「我、汝のコンピュータを祝福す、我が子よ。」

笑いは、数秒後には、満場の喝采に変わる。聴衆が拍手すると、ストールマンの頭上のコンピュータ・ディスクは天井の照明の光を受けて、完璧な光輪の効果を現出させる。ストールマンは、一瞬にして、厄介な本土の白人から、ロシア正教のイコンに変身する。

「始めに Emacs は、テキスト・エディタなりき、」とストールマンは、その来歴を説明して言う。「やがて多くの人々の生き方となり、ある人々の信仰となりき。我らは、この信仰を Emacs 教会と呼ぶ。」

この寸劇は、自分をパロディにした気楽なひとときであり、ストールマン的なソフトウェア上の禁欲主義を、姿を変えた宗教的狂信と見る多くの人々への ユーモラスなジャブの打ち返しだ。それは、もう一方の靴が落ちて立てる大きな音でもある。それは、ローブと光輪を着けたストールマンが、最後に聴衆を解放 して、「笑っていいんですよ。変だってことは分かってます、」と言っているみたいだ。[RMS: 誰かを変てこだと笑うのは野暮だし、それを弁明するのは私の意図ではないが、聖イグヌーティウスという私のお約束のコメディを人々が笑ってくれるのはうれ しい。]

後日、聖イグヌーティウスという登場人物について論じ合ったとき、ストールマンが言うには、それを最初に着想したのは 1996 年で、Emacs をつくってからはかなり経っているが、「オープンソース」という言葉が現れてハッカー・コミュニティの主導権争いをあおり立てるよりは十分前のことだっ た。そのとき、とストールマンは言う、「自分を笑いものにする」ことで、自分は頑固だが、誰かがそう思わせたいと思っているような狂信者ではないことを、 聴衆に思い起こさせる手段が欲しかった。後になってから、とストールマンはつけ加える、「他人が、聖イグヌーティウスをとらえて、ソフトウェア・イデオ ローグだというイメージを強化する方便にしたんです。1999 年の linux.com ウェブサイトのインタビューで、エリック・レイモンドがしたようにね。」

ストールマンが自分の行為を調整しているというとき、彼を貶めたり彼の不誠実さを責

めているわけじゃありません。よい伝達者たちに倣って、彼には演劇的な要素がある、といっているんです。それは、ときに意識的ですが -- 聖イグヌーティウスの衣装を着て、頭に円盤を載せ、ソフトウェアを祝福する彼をみたことがありますか？多くの場合、それは無意識的です。彼は相手を適度に じらしながら刺激しておくすべを心得ていて、(普通は)人を怒らせない範囲で、注意を引きつけます。7

ストールマンはレイモンドの分析に異義を唱える。「これは単に僕が僕自身を楽しむための一つのやり方なんです、」とストールマンは言う。「他の人がこれにそれ以上のものを見るとしても、それは、その人の関心事の投影で、ぼくのものじゃありません。」

こう言ってから、彼は自分がヘボ役者であることを認める。「ふざけてると思います？」とその点についてさらに彼は言う。「僕は注目の的になるのが好 きなんです。」ストールマンによれば、そのために、彼はかつてトーストマスターズの会員になったことがあるらしい。トーストマスターズとは、会員が人前で 話すスキルを向上させるのを助ける団体で、ストールマンは他人にも強く勧めるそうだ。ストールマンは、演劇関係者が羨むほど舞台に上がる機会があり、昔の ボードビリアンたちとのつながりを感じている。マウイ高性能計算センターの講演の数日後に、それとなく、1999 年のリナックス・ワールドでのパフォーマ ンスのことを言って、ストールマンに、グルーチョ・マルクス・コンプレックスがあるか、つまり、自分を会員にするようなクラブには入りたくない、という気 持があるか聞いてみた。8 ストールマンは即座に応じた。「ありません。でも、僕は多くの面でグルーチョ・マルクスを尊敬していて、確かに発言のあるものは、彼の影響を受けていま す。でも、それならハーポの影響だって受けてますよ。」

ストールマンが昔から駄洒落好きだったことからも、グルーチョ・マルクスの影響は明らかだ。とはいえ、駄洒落と言葉遊びはハッカーに共通の特性でも ある。だが、おそらくストールマンの人となりで一番グルーチョ的なのは、駄洒落を繰り出すときに、表情を変えないことだろう。大抵の駄洒落は -- 眉を上げたり、ニヤリとすることでほのめかすこともなく -- 全くさりげなく言われるので、聴衆が彼のことを笑う以上に、ストールマンは聴衆のことを笑っているのでは、と思わずにいられない。

聖イグヌーティウスのパロディを面白がるマウイ高性能計算機センターの人々を見ていると、そんな心配も消えていく。厳密には漫才とはいえないかもし れないが、ストールマンには確かに、部屋中のエンジニアたちを笑いころげさせるだけの技量がある。「Emacs 教会の聖人に列せられるのに必要なのは、禁欲に非ず、道徳的に清い生活を送ることへの誓いなり、」と彼はマウイの聴衆に語りかける。「汝の全てのコン ピュータから、悪なる占有的オペレーティングシステムを追い出し、しかる後に、全てが[聖なる]自由なオペレーティングシステムをインストールすべし。し かる後に、汝、その上に自由なソフトウェアのみをインストールすべし。汝がこの誓いを立て、それにしたがって生きるなら、汝もまた Emacs 教会の聖人となり、光輪を持つであろう。」

聖イグヌーティウスの寸劇は、短い内輪向けのジョークで終る。大方の Unix システム

やその派生物で、Emacs の主要な競争相手は vi（ヴィアイと発音）である。vi は元 UC バークレーの学生で、現在はサン・マイクロシステムズのチーフ・サイエンティストであるビル・ジョイが開発したテキスト・エディタである。その「光輪」を 外す前に、ストールマンはこのライバル・プログラムを少々もてあそぶ。「人はときに、Emacs 教会にて vi を使うことは罪なりや、と問う、」と彼は言う。「自由なバージョンの vi を使うことは罪に非ず。そは苦行なり。なれば、ハッピー・ハッキング。」9

短い質疑応答の後、聴衆の一部がストールマンを取り囲む。中にはサインを求める者もいる。「これにサインしよう、」とストールマンは言い、ある女性 が持っていた GNU 一般公衆ライセンスのプリントを取り上げながら、「でも、あなたが、Linux の代りに GNU/Linux という用語を使い、君の友達全員にも同じようにするように言うと約束するならね。」

このコメントは、私的な観察を確証するものでしかない。他の舞台芸人や政治的人物とは異なり、ストールマンには、「オフ」モードがない。聖イグヌー ティウスのキャラクターは別として、舞台で見るイデオローグと、楽屋で会うイデオローグは同じだ。その晩遅く、夕食の会話の中で、あるプログラマが「オー プンソース」プログラムへの親近感に言及したところ、ストールマンは、モグモグしながら、そのテーブルメートに注意した。「あなたが言っているのは、自由 ソフトウェアのことです。それを言うなら、自由ソフトウェアと言うべきです。」

質疑応答のとき、ストールマンは教育者の役割を引き受けることがある。「よくこう言う人がいます。『そう、まず人々をコミュニティに参加するように 招き、それから彼らに自由について教えよう。』これは筋の通った戦略かもしれませんが、実際には、ほとんど誰もが人々をコミュニティに参加するように招い て、だが、彼らが一度来てしまうと、誰も彼らに自由について教えようなどとはしません。」

この結果は、とストールマンは言う、第三世界の都市と似たところがある。「何百万もの人がやって来ては、スラムを作ります。だが、誰も次の一歩、つ まり彼らをスラムから連れ出すことに取り組もうとしません。ソフトウェアの自由について語ることがよい戦略だと思うなら、次の一歩の取り組みに参加して欲 しい。第一歩のためにしなければならないことも山ほどありますが、次の一歩に取り組む人がもっと必要です。」

「次の一歩」に取り組むとは、自由ソフトウェア運動の根本問題が、受容の問題ではなく、自由の問題だということを納得させることだ。占有的ソフト ウェア産業を内側から改革しようとする人は、無駄骨を折ることになる。「内側からの変革はリスクが大きい、」とストールマンは言う。「ゴルバチョフ並に働 くのでなければ、無力化されてしまうでしょう。」

手が挙がる。ストールマンは、ゴルフシャツを着た一団の一人を指名する。「特許なしで、商業スパイにどうやって対処するんですか？」

「ああ、その二つの問題は、実際には互いに関係ないんです。」とストールマンは言う。

「誰かが他社のソフトを盗もうとしたら、という意味ですけど。」

ストールマンは毒ガススプレーにでも当たったかのようにのけぞる。「ちょっと待ってく

ださい」とストールマンは言う。「盗みですって？ 残念だけど、その発言には大変な先入観があります。ぼくに言えるのは、ぼくはその先入観に組しないということだけです。自由でないソフトウェアやその他の ものを開発している会社は、膨大な企業秘密を持っていて、現状を変えるなんてありそうにありません。昔は -- 1980 年代ですら -- 大部分のプログラマはソフトウェア特許の存在にすら気づいていず、それを気にもしていませんでした。で、何が起こったかと言うと、人々は面白いアイディア を公表し、彼らが自由ソフトウェア運動の中にいないときは細部を秘密にしました。だから、あなたが言っていることに関しては、特許があろうとなかろうと変 わらないんです。」

「でも特許がアイディアの公表に影響しないとしたら、」別の聴衆が飛び入りするが、その声は、話し始めると同時にかき消される。

「いや、特許は影響するんです、」とストールマンは言う。「公表されるということは、そのアイディアが、そのコミュニティのその他全員にとって、 20 年間に渡って、立ち入り禁止になる、ということです。で、それにどんないいことがあるのでしょう？その上、彼らはそれをとても読みにくく書きます。アイ ディアをぼかすと同時に、特許の適用範囲をできるだけ広くするためにです。だから、公表された情報から何かを学ぼうというのは、基本的に無駄なんです。特 許を見ても、自分が何をできないか、という悪い知らせを知るだけです。」

聴衆は静かになる。3 時 15 分に始まった講演は、5 時のホイッスルに近づきつつあり、聞き手の多くは、既に自分の席でそわそわし、週末に向かって飛び出そうと落ち着かない。みんながくたびれて来ているのを 感じとって、ストールマンは部屋を一瞥し、急いで店仕舞いにかかる。最後の質問者を追いたてるために、競売人の「ありませんか、ありませんか、はい売れま した」の調子で、一渡り眺め回した後、「もう終りのようだね、」と彼は言う。誰も手をあげなくなると、ストールマンはいつもの去り際のセリフとともに退場 する。

「ハッピー・ハッキング」と彼は言う。

後注

0. 豆知識（訳者の蛇足とも言う）： イグヌーティウスまたは英語風にイグヌースィアスは、16 世紀のスペインのカトリックの司祭でイエズス会の創設者イグナティウス・デ・ロヨラに、グヌー（GNU）をかけ合わせた駄洒落である。

1 RMS: 皮膚が日照不足になりうるとか、白い肌は「不健康」といった観念は、危険で誤った情報だ。ビタミン D を充分に摂取している限り、太陽にあたらなくても有害ではない。死亡原因にもなる皮膚のダメージになるのは、日光の浴びすぎである。

2 "Grateful Dead Time Capsule: 1985-1995 North American Tour Grosses." http://www.accessplace.com/gdtc/1197.htm を参照。

3 Evan Leibovitch, "Who's Afraid of Big Bad Wolves," ZDNet Tech Update (December 15, 2000), http://www.zdnet.com/news/whos-afraid-of-the-big-bad-wolves/ 298394 を参照。

4 叙述の都合上、私はストールマンによるソフトウェアの「自由」の完全な定義の説明に深入りすることをためらった。GNU プロジェクトのウェブサイトには、四つの基本要素が列挙されている。

プログラムを走行させる自由、目的は問わない(自由 0)。

プログラムがどのように働いているかを調べる自由、また、それを自分の必要に適合させる自由(自由 1)。

隣人の役に立つようにプログラムのコピーを再配布する自由(自由 2)。

プログラムを改良し、改良を公けにリリースし、コミュニティ全体が利益を受けるようにする自由(自由 3)。

より多くの情報を得るには、http://www.gnu.org/philosophy/free-sw.html の "The Free Software Definition" を訪ねられたし。

5 Eric Raymond, "Shut Up and Show Them the Code," online essay, (June 28, 1999). http://www.tuxedo.org/~esr/writings/shut-up-and-show-them.html を参照。

6 同書。

7 "Guest Interview: Eric S. Raymond," Linux.com (May 18, 1999). http://www.linux.com/interviews/19990518/8/ を参照。

8 RMS: ウィリアムズは、心理学的なものとして扱うことにより、グルーチョの有名な言葉を誤解している。それは露骨なユダヤ人差別をしていた多くのクラブの仕事を 言っていた。ユダヤ人差別はクラブがグルーチョを会員にしない理由でもあった。私は、母に教えられるまで、どちらも知らなかった。ウィリアムズや私は、偏 見が地下に葬られてから育った。そして、グルーチョがしたようにユーモアでつつんで婉曲な偏見批判をする必要がない。

9 Emacs 教会のサービスは、2001 年からさらに発展している。今や、ユーザーは信仰告白「GNU のほかにシステムはなし、Linux はそのカーネルの一つに過ぎず。」を暗唱すれば教会に参加できる。ストールマンは、フーバー・ミツバー（the Foobar Mitzvah）として知られる宗教的儀式、Emacs の様々なライバル・ヴァージョン間の大分裂（the Great Schism）、Emacs の処女崇拝（Emacs を使うことをまだ学んでいないすべての人を指している）にしばしば言及している。加うるに、「ヴィアイ、ヴィアイ、ヴィアイ」がエ ディタの獣性を認めるものとなった。

第九章　GNU 一般公衆ライセンス

1985 年の春までに、リチャード・ストールマンは Unix 風オペレーティング・システムのために Lisp 版の Emacs を作った。GNU プロジェクトの最初の実質的成果だ。自由ソフトウェアとして他の人々が利用できるように、それをリリースする方法を開発する必要があった。それは事実上 Emacs コミューンの後継となるものだった。

改良する自由と著作者特権との間の緊張関係は GOSMACS よりも前にできていた。1976 年の著作権法は、アメリカの著作権法を見直し、著作権の法的保護をソフトウェアプログラムにまで拡大した。その法律のセクション 102（b）によれば、今や、個人と企業は「プログラムに具体化した現実のプロセスまたは方法」に対してではなく、ソフトウェアプログラムの「表現」に対 して著作権を持てるようになった。1

言い換えれば、これはプログラムを代数の教科書のように扱うものだった。その著者は教科書の著作権を主張できるが、代数の数学的アイデアやそれを説 明するのに必要な教育的テクニックに対しては著作権を主張できない。かくして、オリジナルの Emacs のコードの使用についてストールマンが何と言っているかと無関係に、他のプログラマには Emacs のアイデアとコマンドについて自分自身の実装を書ける法的資格があり、それを実装した。Gosmacs は、様々なコンピュータ・システムのために開発されたオリジナル Emacs の三十幾つかあった模造品の一つだった。

Emacs コミューンは、ストールマン自身が書いたオリジナル Emacs プログラムのコードにのみ適用された。もしも Emacs コミューンに法的強制力があっても、それは独立して開発された模造品、たとえば Gosmacs には適用されなかっただろう。自由ソフトウェア運動の倫理的理想からすると、自由のない Gosmacs の制作は非倫理的だった。なぜなら、(占有的ソフトウェアとして) それはユーザーの自由に敬意を払わないからだ。しかし、この問題は Gosmacs の作者の理想とは無縁だった。

著作権の下では、既存のプログラムからコードをコピーしたいプログラマは (修正をともなう場合でさえ) オリジナル開発者の許可を手に入れる必要があった。ハッカーはたいてい知らなかったが、著作権表示がなくとも新法は適用された。そして著作権表示もまた現れ始めていた。

これらの表示をストールマンは占領軍の侵略の旗として眺めていた。過去のプログラムからソースコードを借りていないソフトウェアプログラムは稀だっ た。ところがアメリカ政府は、大統領の一筆の署名で、プログラムの再使用を禁じる法的な力をプログラマや会社に与えた。それはこれまで形式ばらないシステ ムだったものの中に、形式的要素を持ち込んだ。簡単に言えば、それまでハッカー間の論争で解決されたものが、今や弁護士間の論争で解決されるものになっ た。そんなシステムは、自動的に、ハッカーではなく会社に有利に働いた。著作権表示に個人名を記載してコードの品質に責任を負うのだと考える者もいたが、 通常、著作権表示には会社名が記載されたし、書いたコードについて個人が発言す

るためには他の方法があった。

しかし、ストールマンは、何年間か GNU プロジェクトを率いているうちに、著作権は、著作権がカバーする一定の活動について寛大な許可を与えることができ、著作権に条件を置くこともできることに 気付いた。「著作権表示つきの電子メールのメッセージに、ライセンスとして『そのままのコピーを許可する』としたものがあったんですが、」とストールマン は回想する。「それでひらめいたんです。」これらのライセンスは、ライセンスの条件を除去しないこと、としていた。ここからストールマンのアイデアまで、 あと数歩だった。たとえば、許諾条件は、ユザーが修正版でさえ再配布することができ、かつ、それらは同じ許諾条件を持つようにすること、とすることが可能 だった。

かくして、ストールマンは、著作権の利用は必ずしも反倫理的ではないと結論づけた。ソフトウェア著作権の駄目な点は、ユーザーに不可欠な自由を認め ない著作権の典型的利用方法と利用法の設計にあったのだ。ほとんどの著作者は著作権を使う別の方法を想像することができなかった。しかし、著作権は別のや り方で使えた。それは、プログラムを自由にして、それが自由であり続けることを保証する利用方法だった。

1985 年の GNU Emacs 16 までに、ストールマンは著作権に基づくライセンスを起草したが、それはユーザーがコピーを作って配布する権利を与えるものだった。それはユーザーが修正版 を作って配布する権利も与えたが、同じライセンスの下でのみ、そうすることできた。ユーザーは修正版に対して無制限の権力を行使することはできなかった。 したがって、修正版を Gosmacs がそうであったように占有的なものにすることはできなかった。そして、ユーザーはソースコードを利用可能にする必要があった。それらの条件は、部外秘的で 自由のない修正版の GNU Emacs を出現させうる法的隙間を閉ざしていた。

Emacs コミューンの社会契約を成文化する役に立ったが、初期の GNU Emacs のライセンスは、GNU プロジェクトが目指すものとしてはまだ「非公式的」過ぎた、とストールマンは言う。自由ソフトウェア財団を設立してまもなく、財団の他の理事や、設立を手 伝ってくれた弁護士と相談しながら、ストールマンは、ライセンスをさらに完全な、乗じる隙のないものにするための作業を開始した。

ボストンの著作権法の弁護士でストールマンに最初に法的アドバイスを提供していたマーク・フィッシャー（Mark Fischer）は、この時期にストールマンと交わしたライセンス論議を次のように回想している。「リチャードは、ライセンスがどんな働きをすべきかということでは非常に強い見解を持ってましてね、」とフィッシャーは言う、「彼には、二つ原則があったんです。一つは、可能な限り、ソフトウェアを絶対的に オープンなものにする、ということ。（これを言ったとき、フィッシャーはオープンソース支持者たちに影響されていたように思う。決してストールマンはソフ トウェアを「オープン」にすることを求めていたのではなかった。）もう一つは、他人にも同じライセンスを採用するように仕向ける、ということです。」ライ センスの要件は第二のゴールのために設計されていた。

この最後の条件の革命性が十分に理解されるまでには、いくらか時を要した。当時、とフ

ィッシャーは言う、GNU Emacs ライセンスを単に普通の取引と考えていた。それは GNU Emacs の使用に値札をつけていた。ストールマンはユーザーに、お金ではなく、彼らが後に加える修正にアクセスできることを要求した。こう言ってフィッ シャーは、契約条項のユニークさを思い起こす。

「他人にこういう対価を受け入れるように求めるのは、唯一じゃないとしても、当時は非常に珍しいことだったと思います」と彼は言う。

GNU Emacs ライセンスを作るにあたって、ストールマンはかつての Emacs コミューンの非公式な主義に大きな変更を一つ加えた。かつてはコミューンメンバーに彼らが書いたらすべての変更を送るように要求したところで、 今やストールマンは、プログラムの再配布を選択したら必ずソースコードと自由を告知することのみを要求した。言い替えれば、自家用の修正を Emacs に加えただけのプログラマは、もうストールマンにソースコードの変更を送り返す必要がなかった。自由ソフトウェアの教義のめったにない変更にあたって、ス トールマンは、自由ソフトウェアの値段を切り下げることにした。どのコピーにも所持者がコピーをさらに開発して再配布する許可が付いている限り、ユーザー は肩越しに覗くストールマンぬきでイノベーションをすることができ、彼らが望むときにだけ彼らの修正版を再配布してよいことになった。

この変更は、初期の Emacs コミューンの社会契約のビッグブラザー的側面に自分自身満足していなかったことが原因だった、とストールマンは言う。 変更を彼に送るのは全員にとって有益だとは分かっていたが、これを要求することは不平等だと感じるようになった。「みんなにすべての変更を発表するように求めるのは間違っていました」とストールマンは言う。

「一人の特権的な開発者に変更を送るように求めるのは間違いでした。そんな集権化と一人のための特権は、全員が平等の権利を持つ社会とは調和しないんです。」

GNU Emacs ライセンスは、1985 年にストールマンがついに GNU Emacs をリリースしたときにデビューした。リリース後、ストールマンはライセンスの文言をどう改善したらいいかに関する一般のハッカーコミュニティからの入力を 歓迎した。この提案を取り上げたハッカーの一人に、後のソフトウェア活動家で、当時サン・マイクロシステムズのコンサルタントとして働いていた、ジョン・ ギルモア（John Gilmore）がいた。ギルモアはコンサルタント業務の一環として、Emacs を Unix の企業内バージョンである SunOS に移植していた。ギルモアはそうする過程で、GNU Emacs ライセンスの要求に従って変更を公表した。ギルモアは、ライセンスを負担と見るのではなく、ハッカー精神の明瞭で簡潔な表現と見ていた。「それまで、ほと んどのライセンスは非常に非公式なものでした」とギルモアは振り返っている。

この非公式ぶりの例として、ギルモアは、Unix のユーティリティ、trn の 1890 年代中ごろの著作権ライセンスを引用する。trn は、後に Unix の patch ユーティリティとスクリプト言語 Perl（パール）の作者として有名になるハッカー、ラリー・ウォール（Larry

Wall）が書いたニュースリーダ・プログラムである。通常のハッカーの礼儀と、商用に公表する方法を指図する作者の権利との均衡をとるために、ウォール は、プログラムに付随する著作権表示を意見表明の手段として利用した。

Copyright (c) 1985, ラリー・ウォール
これで金儲けするとか、これを自分が書いたふりをしない限り、trn キットの全部または一部をコピーできる。2

こういう表現はハッカー倫理の反映だが、倫理のゆるやかで非公式な性質を、著作権という厳密で法律的な言葉に翻訳することの難しさも反映していた。 GNU Emacs ライセンスを書くときに、ストールマンは占有的派生品をつくる抜け道をふさぐ以上のことをした。彼は、法律家にもハッカーにも同じように理解で きる仕方で、ハッカー倫理を表現したのである。

まもなく、とギルモアは言う、他のハッカーが自分のプログラムに GNU Emacs ライセンスを「移植」する方法を議論し始めた。ユーズネットでのやりとりに刺激されて、1986 年 11 月、ギルモアはストールマンに修正を提案するメールを送った。

たぶん、ライセンスから「EMACS」を削除し、「ソフトウェア」とか何か別の名前 で置き換えるべきです。遠からず、と希望しますが、Emacs は GNU システムの一番大きな部分ではなくなり、このライセンスが GNU システム全体に適用されることになるでしょうから。3

もっと一般的なアプローチをするように提案してきたのはギルモアだけではなかった。1986 年の暮までに、ストールマン自身、GNU プロジェクトの次の大きな一里塚であるソースコードデバッガー、GDB にとりかかっていた。このリリースのため、彼は、GNU Emacs にかわって GDB に適用できるように Emacs ライセンスを修正する必要があった。大仕事ではなかったものの、それは間違えるかもしれない仕事の開始だった。1989 年、ストールマンは Emacs への個別的言及を除く方法を解決して、プログラム・コードとプログラムのソースファイルの中にあるライセンスだけを結合することを表明した。このやり方な ら、どのプログラマもライセンスを変更せずに、自分のプログラムにライセンスを適用することができた。GNU 一般公衆ライセンス（The GNU General Public License）、略して GPL の誕生だった。GNU プロジェクトは、まもなくそれを既存の全ての GNU プログラムの公式ライセンスにした。

GPL の公表にあたって、ストールマンは、小さな変更を示すために小数を使い、 大きな変更を示すために整数を使うというソフトウェアの慣習に従った。 初版は、1989 年で、バージョン 1.0 とされた。 このライセンスには、その政治的な狙いを詳しく説明した次のような前文がついている。

GNU 一般公衆ライセンスは、以下のことを保証するるように設計されています。 あなたが自由なソフトウェアのコピーを譲渡または販売する自由を持つこと、 あなたがソースコードを受け取る、または、欲しいときにソースコードを入手できること、あなたが本ソフ

トウェアを変更できる、または、新しい自由なプログラムの中でその一部を使えること、そして、あなたにこれらのことができることを知らせること。 あなたの権利を守るためには、誰かがあなたにこれらの権利を拒否したり、 放棄を求めたりすることを禁ずるように制限をかけることが必要になります。 これらの制限により、本ソフトウェアのコピーを配布するとき、 またはそれを変更するとき、あなたは一定の責任を負うことになります。4

ハックが進むにつれ、GPL はストールマンの最高傑作の一つになった。 それは通常は占有的な著作権法の制約の中に、共有的なシステムを創り出した。 また、さらに重要なのは、法典(リーガル・コード)とソフトウェア・コードの間にある知的類似性を実証したことだ。GPL の前文には、暗黙裏に深いメッセージが込められている。 ハッカーは著作権法をいぶかしいものと見る代りに、それをハックされたがっているもうひとつのシステムと見るべきだ、ということだ。

「GPL は、大コミュニティを持つ自由ソフトウェアによく似た発展をしました。 コミュニティは、ソフトウェアの構造や、賛成意見と反対意見、 また広く支持されるための微調整や穏健に妥協する必要性さえ議論します」と、ジェリー・コーエン（Jerry Cohen）は言う。 彼は、ストールマンがライセンスをつくるのを手伝ったもう一人の弁護士だ。 「このプロセスは非常に良く機能しまして、GPL は、幾つかのバージョンを経ながら、広範囲の懐疑的な、時には敵意ある反応を受けるものから、広範囲の支持を受けるものへと変わっていきました。」

1986 年の Byte 誌とのインタビューで、ストールマンは面白い言葉で GPL を要約した。ハッカーの価値観を賛美することに加えて、ストールマンは言う。 読者は、「それを知的な柔術の型としても見るべきなんです。 ソフトウェアを買い占める人達がつくった法体系を、それに対抗するために使うんですから。」5 後年、ストールマンは、もう少しとげのない言葉で GPL の創造について語っている。 「ぼくは、ある意味で倫理的な、ある意味で政治的な、 ある意味で法律的な問題を考えていました、」と彼は言う。 「ぼくらがその中にいる法体系によって持続できることを試みる必要がありました。 気持ちとしては、この仕事は新しい社会の基礎となる法律を制定することなんですが、 ぼくは国じゃないから、実際に法律を変えることはできません。 こうしたことのために設計されていない既存の法体系の上に増築することで、これを試みる必要がありました。」

自由ソフトウェアに関わる倫理的、政治的、法的問題をストールマンが考えている頃、ドン・ホプキンス（Don Hopkins）というカリフォルニアのハッカーが、彼に 68000 マイクロプロセッサーのマニュアルを郵送してきた。 Unix ハッカーで SF マニア仲間のホプキンスは、 ストールマンからそのマニュアルを借りていたのだ。 ホプキンスは感謝のしるしに地元の SF 大会で入手したステッカーをたくさん貼って返信封筒を飾っていた。そのステッカーの一枚が、ストールマンの目にとまった。 そこには、"Copyleft (L), All Rights Reversed" とあった。 ストールマンはステッカーにヒントを得て、GNU Emacs ライセンスで（そして後に GPL で）使われた法的テクニックに「コピーレフト」という愛称をつけ

た。ユーモラスな、丸に逆向きの C がそのシンボルだ。 時とともに、その愛称は、「あるプログラムを自由ソフトウェアにし、 そのプログラムを修正・拡張した版も同じように自由ソフトウェアにすることを要求する」著作権ライセンスの手法全般を表現する自由ソフトウェア財団の専門用語になった。6

かつてドイツの社会学者、マックスウェーバーは、すべての偉大な宗教はカリスマの「ルーチン化」と「制度化」の上に成り立っていると唱えた。すべての成功した宗教は、 カリスマまたは初代宗教指導者のメッセージを文化や時代を超えて容易に翻訳可能な、社会的、政治的、倫理的な装置に転換する、とウェーバーは論じた。

本来、宗教的なものではないが、GNU GPL は、ソフトウェア開発という現代的で分権的な世界で機能しているこの「ルーチン化」プロセスの興味深い実例に確かにふさわしい。その公開以来、他の点ではストールマンに少しも忠誠心を表明しないプログラマや企業が GPL の取引を額面通りに喜んで受け入れてきた。自身のソフトウェアプログラムの先制的防御メカニズムとして GPL を受け入れる者さえある。GPL の条件を制約が強すぎるとして拒絶する者でさえ、その影響力を認める。

キース・ボスティック（Keith Bostic）は、後者のグループに属するハッカーの一人で、GPL1.0 がリリースされたときはカリフォルニア大学の職員だった。ボスティックの所属部署、コンピュータシステム研究グループ（SRG）は、1970 年代後半から Unix の開発に関わり、現代のインターネット通信の土台になる TCP/IP ネットワークプロトコルを含む、多くの主要部品に責任を持っていた。 1980 年代の終り頃になって、Unix という商標の最初の保有者である AT&T は Unix の商品化に力を入れ始め、ボスティックと彼のバークレーの同僚が開発している Unix のアカデミックバージョン、BSD（バークレー・ソフトウェア・ディストリビューション）を、商業的テクノロジーの主要な源泉として期待し始めていた。

ボスティックと友人達によって書かれたコードは、ほとんど誰にとっても使用禁止状態だった。占有的な AT&T のコードと混ざり合っていたからだ。その結果、バークレー・ディストリビューションは、AT&T から Unix のソースコードライセンスをすでに取得した施設のみが利用できた。 （学問的世界のみを考えている人々にとって）最初は無害に見えたこの仕組みは、AT&T がライセンス料を値上げしたとき、だんだん重荷になった。 バークレーのコードを GNU で使うために、ストールマンは、それを AT&T のコードから切り離し、自由ソフトウェアとしてリリースするようにバークレーを説得する必要があった。1984 年または 1985 年に、彼は奮闘する BSD のリーダーたちと会い、AT&T はチャリティーではなく、大学がその成果を（事実上）AT&T に寄付するのはふさわしくないと指摘した。彼はコードを取り出して自由ソフトウェア としてリリースしてくれるように頼んだ。

1986 年に雇われて以来、ボスティックには、デジタル・イクイップメント社の PDP-11 に BSD を移植するという個人的なプロジェクトがあった。ちょうどこの頃、とボスティックは言う、たまたま西海岸に遠征してきたストールマンと親密に交流するよう になった。

「ストールマンと著作権の議論をしたことを、鮮明に覚えています。ストールマンは、CSRG で借りたワークステーションの前に座っていました、」とボスティックは言う。 「それからディナーに行きましたが、ディナーの最中も著作権について議論を続けました。」

ストールマンの好む形にはならなかったが、その議論は最終的に実を結んだ。1989 年 6 月、バークレーは、そのネットワークコードを AT&T の保有する OS の残り部分から切り離し、著作権をベースにした自由なライセンスで配布し始めた。 そのライセンスの条件は寛大だった。 すなわち、ライセンスをうけるすべての人は、派生的プログラムを広告するときは、 カリフォルニア大学のクレジットを出すこと。6 GPL とは対照的に、占有的な派生物は許可された。 一つの問題だけがライセンスの急速な採用を妨げた。 BSD ネットワーキング・リリースは、完全なオペレーティング・システムではなく、ネットワーク関連の部分に過ぎなかったのだ。 そのコードは、全ての自由なオペレーティング・システムに対する大きな貢献であったにもかかわらず、それを走らせるには、他の占有的なライセンスのコードと一緒にしなければならなかった。

それから数年間、ボスティックと他のカリフォルニア大学の職員は、足りない部品を置き換えて BSD を完全な、自由に再配布可能なオペレーティング・システムにするために働いた。 Unix の商標を管理する AT&T の子会社、ユニックス・システムズ・ラボラトリーが法的な異議を出して遅れたが、努力は、1990 年代の初めに結実した。 しかし、それ以前から、多数のバークレーのユーティリティがストールマンの GNU システムに貢献していた。

ボスティックは回想して言う、「GNU の刺激がなかったら、実際にやったほど力を入れて進めることはなかったと思います。彼らが熱心に活動している領域が重要なのは明白だったし、ぼくらは、そのアイデアが好きだったんです。」

1980 年代終りには、GPL は、自由ソフトウェアのコミュニティに重力効果を及ぼし始めた。BSD ユーティリティの事例に見られるように、自由ソフトウェアと認められるのにライセンスを GPL にする必要はなかったが、プログラムを GPL の下に置けば明確なメッセージが送られることになる。「GPL があったから、みんなが自分たちが作っているのが自由なソフトウェアなのかどうか、どんなライセンスにするかを、とことん考えるようになったと思う。」と ブルース・ペレンス（Bruce Perens）は言う。 彼は、Unix の人気のユーティリティ、エレクトリック・フェンスの作者で、後の Debian GNU/Linux 開発チームのリーダーだ。GPL のリリースから数年後、ストールマンの弁護士審査済みライセンスを支持して、自家製のエレクトリック・フェンスのライセンスは捨てることにしたとペレンスは言う。「それは実際とても簡単でした」とペレンスは回想する。

リッチ・モーリンは、ストールマンが最初に GNU をアナウンスしたときいくらか懐疑的に見ていたプログラマだったが、GPL の傘の下に集まり始めたソフトウェアに強い印象を受けたと回想している。SunOS ユーザーグループのリーダーとして、1980 年代のモーリンの主な任務の一つは、最良のフリーウェアか自由ソフトウェアのユーティリティを収録したディストリビューションテープを発送することだっ た。 その仕事をしていると、プ

ログラムの原作者に、プログラムは著作権で保護されているのか、それともパブリックドメインに委ねられているのか、確認の電話を かけねばならないことがしばしばだった。1989年頃、一番優れたソフトウェアプログラムはいつも決まって GPL のライセンスを採用していることに気付き始めたとモーリンは言う。「ソフトウェアのディストリビュータとして、GPL という言葉を見たら、これは楽勝だとわかりました。」とモーリンは回想する。

Sun ユーザーグループのためにディストリビューションテープをまとめる手間の埋め合わせに、モーリンは受取人に手数料を請求していた。だが、GPL を採用したプログラムの場合には、モーリンは突然以前の半分くらいの時間でテープをまとめられるようになり、かなりのもうけが出るようになった。 商機を感じて、モーリンは自分の趣味にビジネスとしての新しい名前をつけることにした。名付けて、プライムタイム・フリーウェアと言う。

そういう営利的利用は、自由ソフトウェアの予定範囲内にある。「私たちがフリーなソフトウェアと言うとき、それは自由（フリーダム）について語っているのであって、価格のことではありません。」GPL の前文の中でストールマンはそうアドバイスした。1980 年代の終わり頃、ストールマンはそれをもっと簡単で覚えやすい形に洗練した。「フリービール（無料のビール）のフリーじゃなくて、 フリースピーチ（言論の自由）のフリーだと思ってください。」

大部分の企業は、ストールマンの嘆願を無視した。 しかし、何人かの起業者にとって、自由ソフトウェアが意味する自由は、 自由市場の意味の自由だった。ソフトウェア所有権を商業的方程式からはずせば、 弱小なソフトウェア会社でも、世界の IBM や DEC と自由に競争できる状況が生まれた。

この考え方を理解した最初の起業者の一人がマイケル・ティーマン（Michael Tiemann）だ。彼はソフトウェアプログラマでスタンフォード大学の大学院生だった。 1980 年代のティーマンは、お気に入りのアーチストの後に続こうとする野心的なジャズミュージシャンのように GNU プロジェクトについていった。しかし、1987 年に GNU C コンパイラつまり GCC がリリースされる頃には、 自由ソフトウェアの潜在力を全面的に理解し始めていた。 GCC を「爆弾」と呼ぶティーマンは、 このプログラムの存在自体がストールマンのプログラマとしての決意を物語っていたと言う。

「ちょうど、どの作家も偉大なアメリカの小説を書くことを夢見るように、 1980 年代のプログラマは偉大なアメリカのコンパイラを書くことについて語っていたんだ。」とティーマンは回想する。「突然、ストールマンはそれをやってしまった。恐れ入った。」

「君は、一つだけ欠けていた物の話をしているけど、GCC がそれだった。」 とボスティックも繰り返す。「当時、GCC ができるまでは、誰もコンパイラなんて持ってなかったんだ。」

ティーマンはストールマンと競うより、彼の仕事の上に増築することに決めた。GCC のオリジナルバージョンのコードは、110,000 行もあったが、 プログラムは驚くほど理解しやすいものだった、とティーマンは回想する。 事実、とてもやさしかったので、5 日足ら

ずでマスターし、 もう一週間かけてソフトウェアを新しいハードウェアプラットフォーム、ナショナル・セミコンダクターの 32032 マイクロチップに移植した、 とティーマンは言う。 それから一年間、C と並んで C++ も扱えるように GCC を拡張することで、C++ プログラム言語の「ネイティブ」な、つまり直接的なコンパイラをつくろうとして、 ティーマンはソースコードをいじっていた。(既存の C++ 言語の占有的実装は、C 言語に変換してから、次に、その結果を C コンパイラに食わせるものだった。) ある日ティーマンは、ベル研でこのプログラムについてレクチャーをしたとき、 同じ事を達成するために奮闘している AT&T の開発者に出会った。

「部屋には、四、五十人の人がいて、何人がネイティブコードのコンパイラに取り組んでいるのか聞いてみたんだ。」とティーマンは回想する。「主催者は、その情報は秘密だけど、部屋を見渡せばある程度のことはわかるだろう、って言うんだ。」

頭の中で電球がぱっと明るくなるのにそう時間はかからなかった、とティーマンは言う。「六ヶ月間そのプロジェクトに取り組んでいたところだった、」とティーマンは言う。「密かに思ったんだ。私かコードは、自由な市場が報酬をくれる能力のレベルまで来てるぞって。」

ティーマンは、GNU マニフェストの中にさらなるインスピレーションを感じた。 マニフェストは占有的ソフトウェアベンダーの貪欲を激しく非難する一方で、会社が、 消費者の視点から自由ソフトウェアの利点を考え、企業活動にあたって自由ソフトウェアを使用し、再配布することを奨めていた。GPL は、商業的ソフトウェアの問題から独占の力を取り除くことによって、 最もスマートなベンダーがサービスに基づいて競争することを可能にする。サービスは単純な技術サポートから個別の依頼者の必要に応じた自由なプログラムの拡張のトレーニングまで広がっている。

1999 年のエッセーで、ティーマンはストールマンのマニフェストの衝撃を回想する。「それは社会主義者の議論のようにも読めるが、私は別の何かを見ていた。 変装したビジネスプランを見ていたのだ。」7

このビジネスプランは新しいものではなかった。80 年代後半に、ストールマンは小規模にこれをやることで自分を支えていた。しかし、ティーマンはそれを新しいレベルに持って行こうとしていた。ティーマンは ジョン・ギルモアとデヴィッド・ ウォリス（David Vinayak Wallace）とチームを組んで、GNU プログラムのカスタマイズに特化したソフトウェアのコンサルティングサービスを立ち上げた。シグナスサポート（Cygnus Support）と称するその会社が、最初の開発契約にサインしたのは 1990 年 2 月のことだ。その年の暮には、会社はサポートと開発契約で 725,000 ドル相当の財産を手にしていた。(シグナスは、非公式的に、"Cygnus, Your GNU Support" の再帰的頭字語である。)

ストールマンが構想した完全な GNU オペレーティング・システムはソフトウェア開発ツール以上のものを必要としていた。1990 年代に、GNU はコマンドライン・インタープリター、すなわち、「シェル（Shell)」も開発した。それはボーン・シェルを拡大し、置き

換えたものだ（FSF の職員のブライアン・フォックス（Brian Fox）が書き、ストールマンが Bourne Again Shell すなわち BASH と命名）。PostScript? のインタープリターの Ghostscript、文書ブラウザ・プラットフォームの Texinfo、プログラムを走らせ、システム・カーネルとやりとりするために C 言語のプログラムが必要とする C Library、表計算ソフトの Oleo（「高価な表計算ソフトより健康志向の」）があり、さらに、かなり良いチェス・ゲームさえあった。しかし、たいていプログラマは、GNU プログラミング・ツールに最も関心があった。

GNU Emacs、GDB、GCC は、開発者向けツールの「ビッグスリー」だが、GNU プロジェクトが 80 年代に開発したのはそれだけではない。1990 年までに、ビルド・コントローラーの Make、パーサー・ジェネレーターの YACC（Bison と改名）、awk（gawk と改名）も作った。 GCC と同様、どの GNU プログラムも、一つのベンダーのプラットフォームだけでなく、様々なシステムで走るように設計されねばならなかった。 プログラムをより柔軟にしていく過程で、ストールマンと彼の協力者たちはしばしばそれらをより便利なものにした。

GNU の普遍主義的アプローチを回想して、プライムタイム・フリーウェアのモーリンは、hello という平凡だが重要なソフトウェアパッケージを指摘する。「C で書かれた 5 行の hello world プログラムなんですけど、 まるで GNU ディストリビューションみたいにパッケージングされていたんです」 とモーリンは言う。 「だから、それには Texinfo や configure の材料があり、 パッケージを様々な環境にスムーズに移植できるようにするために GNU プロジェクトが提供するその他のソフトウェアエンジニアリング用の付随物を備えていました。それは途方もなく重要な仕事で、すべての[ストールマンの]ソ フトウェアに影響しているだけでなく、GNU プロジェクトの他のすべてのソフトウェアにも影響しています。」

ストールマンによれば、ソフトウェアプログラムをつくることが第一で、 改良は二義的なことだった。 「各構成要素について、ぼくはそれを改良する方法を見つけることもあるでしょうし、 見つけないこともあるでしょう」とストールマンは Byte 誌に語っている。「ぼくは、ある程度まで再実装から利益を得ています。再実装すると大抵のシステムはずっと良くなるんです。それは、ある程度まで、ぼくがこの分野に長いこ といて、他の多くのシステムでも仕事をしてきたからだと思います。だから、ぼくは沢山のアイデアを持ち込めるんです。」8

だが、1980 年代の終わり頃に GNU ツールが有名になるにつれてまもなく、AI ラボ仕込みのストールマンの設計の入念さの評判は、あらゆるソフトウェア開発コミュニティの伝説になった。

1980 年代の終わり頃の Sun ユーザーで、1990 年代には自身の自由ソフトウェアのプロジェクト、Samba を運営することになるジェレミー・アリソンは笑いながらその評判を回想してくれた。アリソンは、1980 年代の終わり頃に Emacs を使い始めた。 コミュニ

ティ開発モデルに刺激されて、とアリソンは言う、ソースコードの断片を送ったけれども、ストールマンに却下されただけだった。

「Onion の見出しみたいだったね」とアリソンは言う。「『子供、神に祈るが答えはノー』ってね。」

GNU プロジェクトがユーザーレベルのプログラムとライブラリの創造で成功から成功へと活躍していたときに、カーネルの開発は遅延していた。カーネルは、他のプ ログラムのプロセッサへのアクセスとマシンのリソース全体へのアクセスをコントロールする「交通警官」プログラムだ。

他の主なシステム・コンポーネントの幾つかがそうであるように、ストールマンは修正するための既存のプログラムを探すことからカーネル開発を始め た。1980 年代末の GNU プロジェクトの評論誌「グニュース・レターズ」は、Pastel から GCC を造ろうとした初期の試みのように、このアプローチは最善ではなかったことを示している。1987 年 1 月のグニュースレターによれば、GNU プロジェクトはすでに MIT で開発された Unix カーネル、TRIX のオーバーホールに取り組んでいた。しかし、ストールマンは実際のところ、これを試みようとしていなかった。GCC をやっているところだったからだ。後に、彼は TRIX で良い出発点を得るには余りに多くの変更を必要をとしすぎると結論づけた。1988 年 2 月までに、その月のニュースレターによると、GNU プロジェクトは、カーネギーメロン大学で開発された軽量の「マイクロカーネル」、Mach（マーク）に関心を移したとアナウンスした。Mach は自由ソフトウェアではなかった。しかし、その開発者たちは、非公式に、それを自由なものにすると言っていた。そうなったのは 1990 年で、オフィシャルな GNU プロジェクトのカーネル開発が現実に開始された。9

カーネル開発の遅れは、この時期、ストールマンにのしかかっていた多くの懸念の一つに過ぎなかった。1989 年、ロータス・デベロップメント・コーポレーションは、 人気の表計算プログラム、ロータス 1-2-3 のメニューコマンドを模倣したという理由で、 ライバルのソフトウェア会社、ペーパーバック・ソフトウェア・インターナショナルに対する訴訟を起こした。ロータスの訴訟は、アップルとマイクロソフトの 「ルックアンドフィール」の戦いとともに、GNU システムの未来を危険なものにしていた。どちらの訴訟も GNU プロジェクトを直接に攻撃しているものではなかったが、多くの GNU プログラムがそうであるような、既存のプログラムと互換性のあるプログラムを開発する権利を脅かしていた。これらの訴訟はソフトウェア開発文化全体にぞっ とする効果（chilling effect）を及ぼしかねなかった。 何とかすべきだと決心したストールマンと数人の教授は The Tech（MIT の学生新聞）に広告を載せ、その訴訟を非難して、ロータスとアップルのボイコットを呼びかけた。 彼は、訴訟をしている会社に抗議するグループの組織化に手を貸して、広告後の行動を支援した。 プログラミングの自由同盟と名乗ったそのグループは、 ロータスのオフィスの外で抗議行動を行った。

これらの抗議行動は注目に値した。10

これらはソフトウェア産業が進化していく性質を持っていることの記録になっている。企業の主戦場は、いつのまにかオペレーティング・システムからアプリケーションに置き換っていた。 自由なソフトウェアのオペレーティングシステムを造ることを追求しながら、未だに実現されていないために、流行と成功が一番の価値観になっている人々からすれば、GNU プロジェクトは絶望的に時代遅れに見えた。 ストールマンが「ルックアンドフィール」訴訟との戦いに献身するまったく新しいグループを集める必要性を感じている事実そのものが、一部の観察者の目には FSF は時代遅れだと考えさせるものに映った。

しかしながら、ストールマンには戦略的な理由があった。新しい独占企業が押し付けてきたソフトウェア開発に対するペテンには別々の組織で戦うのだ。 占有的なソフトウェアの開発者もそれに加わるだろう。著作権をインターフェースをカバーするものに拡大することは、多くの自由ソフトウェア開発者と並んで 多くの占有的ソフトウェア開発者を脅威にさらす。これらの占有的ソフトウェア開発者は自由ソフトウェア財団を支持することを望まないだろうが、プログラミ ングの自由同盟には、ことさら彼らを追い払うようなものは何もなかった。同じ理由から、ストールマンはプログラミングの自由同盟のリーダーシップをそれが ものになりそうになるとすぐに手放した。

1990 年に、マッカーサー財団 (the John D. and Catherine T. MacArthur? Foundation) がストールマンにマッカーサー・フェローシップを与えることで、彼を天才と認定したことから、ストールマンはいわゆる「ジーニアス (天才)助成金」を受け取ることになったが、この場合には 5 年間で総計 24 万ドルだった。財団は受賞理由の声明を出さなかったが、これは GNU プロジェクトを立ち上げ、自由ソフトウェアの哲学を提唱したことに贈られたものとして理解された。賞金はストールマンを当面の幾つかの心配事から解放し た。たとえば、それは 80 年代に彼の収入源だったコンサルティングの仕事をやめて自由ソフトウェアの大義にもっと専念することを可能にした。

受賞はストールマンが正常に投票登録できるようにもしてくれた。ストールマンは 1985 年に住んでいた家が火事にあって公式の住所を失ったままになっていた。また、火事は彼の本も大部分灰にしてしまったが、これらの「不潔な本」の掃除も満足 な結果を与えるものではなかった。そのとき以来、ストールマンはテクノロジー・スクエア 545 の「不動産占有者 (squatter)」として生活していたが、投票は「家のない人 (homeless person)」としてせざるを得なかった。11「[ケンブリッジ選挙委員会は]それをぼくの住所と認めたがらなかったんです。」とストールマンは回想し ている。「マッカーサー助成金の新聞記事がそのことを書いたら、ぼくは登録されたんです。」12

最も重要なことは、マッカーサー・フェローシップは、ストールマンにプレスの関心と講演の招待を与えたことで、彼はそれを使い、GNU と自由ソフトウェア、「ルック・アンド・フィール」訴訟やソフトウェア特許の危険のニュースを広めるために使った。

興味深いことに、GNU システムの完成は、これらの旅行の一つに端を発している。1991 年の 4 月にストールマンは、フィンランドのヘルシンキ科学技術大学を訪れた。聴衆の中

には 21 歳のリーナス・トーバルズがいた。彼は、リナックス・カーネルを開発し始めようとしていた。GNU システムに残っていた最も大きな隙間を埋める運命にある自由ソフトウェアのカーネルだ。

当時、すぐそばにあるヘルシンキ大学の学生だったトーバルズは、ストールマンをぼんやり眺めていた。2001 年の自伝『それがぼくには楽しかったから』の中で、トーバルズは、「ぼくは初めて、型どおりに長髪で髭をはやしたハッカーの典型を見た。ヘルシンキにはあまりいない人だった。」と回想している。13

ストールマンの議題の「社会政治的な」側面はピンとこなかったが、トーバルズは議題の基底をなす論理、エラーのないコードを書くプログラマなんてい ない、を高く評価した。特定の好みに合わせてプログラムを修正するユーザーの要望がないときでさえ、いかなるプログラムも改良できるのである。 ソフトウェアを共有することで、ハッカーは貪欲やエゴの防衛といった個人的な動機より、プログラムの改良を優先するのだ。

トーバルズは同世代の多くのプログラマと同様、IBM 7094 のようなメインフレームコンピュータではなく、雑多な取り合わせの自家製コンピュータで爪を研いでいた。 トーバルズは、大学の学生として、大学のマイクロ VAX を使いながら、PC プログラミングから Unix プログラミングへと進んでいった。この段階的前進は、マシンアクセスへの障害に関してトーバルズに別の展望を与えた。ストールマンにとって、主な障害は官 僚主義と特権だった。トーバルズにとって、主な障害は地理とヘルシンキの厳冬だった。 自分の Unix のアカウントにログインしたいだけなのに、ヘルシンキ大学まではつらい道のりを歩かないといけなかった。 そこですぐ、トーバルズはキャンパスから離れたアパートの暖かい部屋からログインする方法を探し始めた。

トーバルズは Minix（ミニックス）を使っていたが、それはオランダの大学教授、アンドリュー・タネンバウム（Andrew Tanenbaum）が教育的手本として開発した自由のない軽量システムだった。14 それは自由のない自由大学コンパイラ・キットに加えて、タネンバウムが 1983 年の手紙でストールマンを軽蔑的に招待した類いのユーティリティを含んでいた。15

Minix は、トーバルズの 386PC のメモリーの限界内に収まったが、実用というより教材用だった。Minix システムは、ターミナル・エミュレーションの装備も欠けていたが、ターミナル・エミュレーションは典型的なディスプレイ・ターミナルをまねして、自宅から トーバルズがマイクロ VAX にログインすることを可能にするものだった。

1991 年初頭に、トーバルズはターミナル・エミュレーションのプログラムを書き始めた。彼は自分のエミュレータの開発に Minix を使ったが、エミュレータは Minix では走らなかった。それは、スタンドアローン・プログラムだった。次に、彼はそれに Minix のファイルシステムにあるディスク・ファイルにアクセスする機能を与えた。その頃、トーバルズは彼の進化していく作品を「GNU/Emacs のターミナル・エミュレーション・プログラム」と呼んでいた。16

Minix が多くの重要な機能を欠いていたので、トーバルズは、彼のターミナル・エミュレータを Minix のカーネルにカーネル互換性を持つものへと拡張し始めた。ただ、モノリシックにした点は別だった。彼は意欲的に、Minix のニュースグループに、Unix 互換カ−ネルの標準仕様書、POSIX（ポジックス）規格のコピーを求めた。17 数週間後、カーネルと GNU プログラムの幾つかをまとめ、それらが一緒に動作するように調整し、トーバルズはメッセージをポストした。それは 1983 年のストールマンの最初の GNU のポストを連想させる。

minix を使っているみなさん、こんにちは。

いま 386(486) 互換機用に（自由な）オペレーティング・システムを作っています （ただの趣味なので、GNU のように大きく、専門的なものにはならないでしょう）。 四月から暖めてきましたが、ようやく準備ができてきたところです。 minix の好きなところ／嫌いなところを教えてもらえませんか。 というのは、ぼくの OS は minix にいくらか似ているからです （とくに（ごく実際的な理由から）ファイルシステムの物理的なレイアウトは同じです)。bash (1.08)と gcc (1.40)を現在移植中です。. . . 18

この投稿が少しばかり反応を引き出したので、一月たたないうちにトーバルズは、 オペレーティング・システムのバージョン 0.01、 すなわち、ありうる最も初期段階の外部レビュー可能なバージョンを、 インターネットの FTP サイトにポストした。 この過程で、トーバルズはこの新しいシステムに名前をつけなければならなくなった。 トーバルズは、自分のパソコンのハードディスク上には、 このプログラムを Linux（リナックス -- 現在普及している英語的な発音。もとの発音に近いのはリヌックス）という名前で保存していた。Unix の変種に名前を与えるときのしきたりを尊重して末尾に X をつけていた。 この名前は「自己中心的」すぎると思って、トーバルズはフリークス（Freax）という名前に変えたのだったが、FTP サイトの管理者に元の名前に戻されてしまった。

トーバルズが、自由なオペレーティング・システムを書いているところだと言い、それを GNU と対比していたことは、彼が完成したシステムを意味していたことを示している。しかし、彼が実際に書いていたものは純然たるカーネルだった。トーバルズは カーネルのほかにはそれ以上書く必要がなかった。なぜなら、彼が知っていたように、他に必要なコンポーネントは GNU や他の自由ソフトウェア・プロジェクトの開発者のおかげで、すでに利用可能になっていたからだった。GNU プロジェクトが GNU システムのためにそれら全てを使うことを望んで以来、それらは必然的に一緒に動作した。トーバルズがカーネルの開発を続けていた間、彼（と後の彼の協力 者）もまたそれらのプログラムを動作させた。

当初、Linux は自由ソフトウェアではなかった。そのライセンスは自由の要件を満たさなかった。商業的配布を認めていなかったからである。トーバルズは企業が急襲して彼 から Linux を取り上げてしまうんじゃないかと心配していた。しかし、トーバルズは GNU/Linux の組み合わせが人気を獲得するにつれ、コピーの販売はコミュニティに有益だと考えるようになった。そして、乗っ取りに対する懸念は心配するまでもないと感 じ始め

た。19 そこで、Linux のライセンスを考え直すことにした。

Linux を GCC でコンパイルしても、Linux で GCC を走らせても、GNU GPL で Linux をリリースする法的必要性はなかったが、トーバルズが GCC を使っていることは、他のユーザーにそれを借用させる道徳的責任を含意していた。トーバルズが後に述べたように、「ぼくは巨人の肩の上に乗っていた」の だった。20 彼が、他の人が同じような支援のために彼をあてにしてきたときに何が起きるか考え始めたのは驚くに当たらない。 決断の十年後に、トーバルズは自由ソフトウェア財団のロバート・シャッセルに、 当時の彼の考え方を要約して鸚鵡返ししている。

人生の半年をこれに費やし、それを使えるものにし、 何か重要なものをそれから得ようとしたけれど、 他人はそれに便乗するだけ、というようにはしたくなかった。 ぼくは[Linux を]みんなに見えるようにして、 人々が心に描く内容に合わせてそれを変更したり改良したりしようと思いました。 でも、みんながしていることも、ちゃんとわかるようにしたかったんです。 自分がいつでもソースにアクセスできるようにしておいて、 誰かが改良を加えたら、それを自分のものにできるようにしたかったんです。21

リナックスのバージョン 0.12 は、初めて完全に統合された GCC を含むバージョンだったが、そのリリースのとき、トーバルズは自由ソフトウェア運動への忠誠を発言することに決めた。彼は古いカーネルライセンスを捨て、 それを GPL に置き換えた。3 年たたないうちに、Linux の開発者はカーネルの Linux 1.0 をリリースした。カーネルは、GNU プロジェクトその他からのプログラムで構成された殆ど完成した GNU システムとともにスムーズに動作した。事実、Linux を加えたことにより、Linux の開発者は、GNU オペレーティング・システムを完成させていたのだった。出来上がったシステムは基本的に GNU プラス Linux だった。しかしながら、トーバルズと彼の友人達は、それを混同を引き起こす Linux として言及した。

1994 年には、この寄せ集めのオペレーティングシステムはハッカー界で十分な尊敬を勝ち得ていたが、観察者の中には、プロジェクトの最初の数ヶ月のうちにトーバ ルズが GPL に切り替えることで、農園を手放すようなことをしていなかったらどうなっていただろう、と思う者もいた。リナックスジャーナルの創刊号で発行者のロバー ト・ヤングは、トーバルズとのインタビューの席についていた。ヤングがこのフィンランド人プログラマに、リナックスのソースコードの私的所有権を放棄した ことを後悔していないのかと聞くと、トーバルズはしていないと答えた。 「とても賢い後知恵で考えても、そうです」とトーバルズは言った。 リナックスのプロジェクトが初期段階で下した「最良の設計上の意志決定の一つ」が GPL だったと彼は考えていた。22

この決定が、ストールマンや自由ソフトウェア財団に全くアピールせず、関心もひかなかったことは、GPL の広範な普及を物語っている。ストールマンに認識されるまでに数年かかったが、リナックス開発の爆発性は、かつての Emacs を彷彿とさせる。 しかし、今回、爆発を引き起こした革新は Control-R のようなソフトウェアハックではなく、PC アーキ

テクチャ上で Unix 風システムを走行させることの新規性だ。動機は違うかもしれないが、最終結果は倫理上の仕様に合致している。 すなわち、すべて自由ソフトウェアで構成された完全に機能するオペレーティング・システムである。

comp.os.minix ニュースグループへの最初のメールメッセージが示すように、トーバルズがリナックスを、GNU 開発者が HURD カーネルを提供するまでのつなぎ以上の何物かだと認識するまで、数ヶ月を要した。トーバルズに関する限り、彼は、単に楽しいだけで分解しては組み立てる子 供たちの長い系譜の最近の一人にすぎなかった。 とはいえ、打ち捨てられたコンピュータのハードドライブの上で余生を送ることになってもおかしくなかったプロジェクトが、 とてつもない成功を収めたことを総括して、 トーバルズは、支配権を放棄し、 GPL という取引を受け入れる知恵を持っていた若き日の自分を評価する。

1991 年のストールマンの科学技術大学での講演と、それに続く GPL への切り替えという決定を回想してトーバルズは次のように書いている。「ちゃんと理解してはいなかったかもしれないが、少なくとも彼の話の一部は頭の中に染み込んだんだと思う。」23

後注

1 Hal Abelson, Mike Fischer, and Joanne Costello, "Software and Copyright Law," 改訂版（1998）を参照。

http://groups.csail.mit.edu/mac/classes/ 6.805/articles/int-prop/software-copyright.html

2. trn キットの README を参照。

http://stuff.mit.edu/afs/sipb/project/trn/src/ trn-3.6/README

3. ジョン・ギルモアの著者への e メールから引用。

4 リチャード・ストールマン他、『GNU 一般公衆ライセンス 第 1 版』（1989 年 2 月）を参照。

http://www.gnu.org/licenses/old-licenses/gpl-1.0.html

5 デイビット・ベッツ、ジョン・エドワード「リチャード・ストールマンがバイト誌編集者と彼のパブリック・ドメインの[原文のまま] Unix 互換ソフトウェアシステムについて語る」BYTE 誌（1996 年 7 月）参照。

(GNU プロジェクトのウェブサイトに再録。 http://www.gnu.org/gnu/byte-interview.html)

このインタビューは、GNU プロジェクトを始めた頃のストールマンの政治姿勢について、率直さはもちろんだが、興味深いものを垣間見せてくれる。 ストールマンの言い回しの変化を跡付けるためにも有益だ。

GPL の目的を説明するために、ストールマンは次のように言っている。「ぼくは、知識と情報一般に人がアプローチする方法を変えようとしている。知識を私物化し ようとすること、人が知識を使っていいかどうかコントロールしようとすること、他人が知識を共有しようとするのをやめさせること、これらは生産妨害だと思 う。」

2000 年 8 月の著者への次の声明とこれを対比されたい。「あなたの思考の中で『知的財産』という用語を使わないように説得したい。それは事態を誤解するように導くでしょう。なぜなら、その用語は、著作権、特許権、商標権を一緒くたにしているからです。 これらの効果は非常に異なっています。 それらを同時に語ろうとするのは全く馬鹿げています。もし誰かが知的財産について、カッコ付きじゃなくて何か言っているときは、 その人は不明瞭に考えているから唱和すべきじゃないのです。」

［RMS: その対比が示すものは、ぼくが一般化に対して、より注意深くなることを学んだということだ。今日では、おそらくぼくは「知識の私物化」について語らないだ ろう、なぜなら、それは非常に広い観念だからだ。しかし「知識の私物化」は「知的財産」と同じような一般化ではない。これらの三つの法律の間の違いは、知 識の私物化についてのいかなる法的問題の理解にも、決定的だ。特許権は一定の知識を使うことへの直接的独占だ。それは実際に「知識の私物化」の一形態だ。 著作権は知識を体現する、または解説する作品の共有を止めるために使われる手法の一つで、全く違った事柄だ。他方、商標は知識という主題とはほとんど関係 がない。］

6 カリフォルニア大学の「うざったい広告条項」は、その後、 問題があることが明らかになった。 ハッカーの中には、GPL よりも穏やかな制限を求めて、 カリフォルニア大学のライセンスを「カリフォルニア大学」というところだけ自分の組織の名前に置き換えて使っていた者もいた。 その結果、他のプログラムから数十ヶ所借りているプログラムは、 数十ヶ所の組織名を広告で引用する必要があった。 1999 年、ストールマン側の十年がかりの働きかけを経て、 カリフォルニア大学はこの条項を削ることに同意した。

BSD ライセンスが抱える問題参照。http://www.gnu.org/philosophy/bsd.ja.html

7 マイケル・ティーマン、「シグナスソリューションズ社の将来性」『オープンソース』（1999 年、オライリー・ジャパン）143 ページを参照。

8 リチャード・ストールマン、BYTE 誌（1986 年）参照。

9『HURD の歴史』を参照。

http://www.gnu.org/software/hurd/history.html

10 プログラミングの自由同盟プレスリリースによれば、 この抗議行動は、十六進プロテスト数え唄が始めて歌われたことで著名である。

1-2-3-4、弁護士はドアから外に放り出せ
5-6-7-8、発明しろ（innovate）、訴訟（litigate）をするな
9-A-B-C、1-2-3 はいらないぞ
D-E-F-O、ルックアンドフィールはもうお仕舞い

http:// progfree.org/Links/prep.ai.mit.edu/demo.final.release

11 Reuven Lerner, "Stallman wins $240,000 MacArthur?? award," MIT, The Tech (July 18, 1990). http://the-tech.mit.edu/V110/N30/rms.30n.html を参照。

12 マイケル・グロス、「リチャード・ストールマン。高校の不適応者、自由ソフトウェア

のシンボル、マッカーサー奨学金の天才」(1999 年) を参照。

13 リーナス・トーバルズ、デイビッド・ダイアモンド共著「それがぼくには楽しかったから」(小学館プロダクション、2001 年) 99 ページ参照。この本はおそらく、トーバルズの人生に関しては正確だが、ストールマンについてはしばしば誤ったことを言っている、たとえば、ストールマンは 「すべてをオープンソースにしたがっている、」そして「他の人が GPL を使わないことに不平を言っている。」と言う。実際には、ストールマンが擁護しているのは自由ソフトウェアで、オープンソースではない。彼は、ほとんどの 環境で、作者に GNU GPL を選ぶように勧めるが、全ての自由ソフトウェアを倫理的だと言っている。

14 1991 年には自由なものではなかったが、今日、Minix は自由ソフトウェアである。

15 タネンバウムは、彼の本、『オペレーティング・システム設計と実装』で Minix を「オペレーティング・システム」として記述しているが、本の議論は Unix のカーネルに対応したシステムの部分のみである。「オペレーティング・システム」という用語の慣習的用法は二つあり、その一つが Unix テクノロジーで「カーネル」と呼ばれるものだ。しかし、それだけが主題における用語的混乱ではない。Minix のその部分は、マイクロカーネル・プラス・その上で走るサーバー群から成っており、GNU Hurd・プラス・Mach と同じ種類の設計だ。マイクロカーネル・プラス・サーバーは、Unix のカーネルに相当する。しかし、その本はカーネルと言うとき、マイクロカーネルにしか言及していない。Andrew Tanenbaum, Operating System Design and Implementation, 1987 を参照。

16 リーナス・トーバルズ、デイビッド・ダイアモンド共著「それがぼくには楽しかったから」(小学館プロダクション、2001 年) 128 頁参照。

17 POSIX は、その後に、多くのコマンドライン機能の仕様書を含むものに拡大されたが、それは 1991 年には存在しなかった。

18 前掲書、138 頁

19 前掲書、153-155 頁

20 前掲書、154 頁

21 前掲書、152-153 頁

22 ロバート・ヤング「リナックス作成者リーナスとのインタビュー」リナックス・ジャーナル(1994 年 3 月 1 日号)を参照。

http://www.linuxjournal.com/article/2736

23 前掲書、99 頁

第十章　GNU/Linux

1993 年までに、自由ソフトウェア運動は岐路に立っていた。楽観的な傾向の人々には、あらゆる兆候がハッカー文化の成功を示していた。データの暗号化やユーズ ネットやソフトウェアの自由について記事を提供するファンキーで新しい出版物のワイアード・マガジンが、雑誌棚から飛び立とうとしていた。かつて、ハッ カーや科学研究者しか使わないスラングだったインターネットは、日常用語の仲間入りをした。クリントン大統領でさえインターネットを使っていた。かつて、 趣味人のおもちゃだったパーソナル・コンピュータは、社会的地位を完全に認知されるまでに成長し、ハッカーが作ったソフトウェアにアクセスするコンピュー タ・ユーザーのまったく新しい一世代を生み出した。そして、GNU プロジェクトは未だに完全に自由なソフトウェアのオペレーティング・システムという目標にたどり着ついていなかったが、すでにユーザーは、変種の GNU/Linux を走らせることができた。

どこから見ても、聞こえて来るのは良いニュースだった。少なくともそう思えた。10 年の奮闘の末、ハッカーとその価値観は、ついに一般社会で受け容れられた。人々がそこにたどり着いたのだ。

本当にそうだったのだろうか？悲観的な傾向の人々にとって、受容を示すどの兆候も、それぞれ固有のトラブルの兆候をともなっていた。たしかに、ハッ カーであることは突然クールなことになったが、うとまれながら育ってきたコミュニティにとって、クールは良いことなのだろうか？たしかに、ホワイトハウス はインターネットについて全く結構なことを言い、自らのドメイン名 whitehouse.gov まで登録したが、それはまた、フロンティア時代の西部のようなインターネット文化を手なずけようとする会社、検閲の味方、法執行機関との遭遇でもあった。 たしかに、パソコンは一層パワフルになったが、自社のチップを搭載したパソコンを市場の定番にしていく過程で、インテルは、今や占有的ソフトウェアのベンダーが力を振るう状況を作り出していた。Linux によって自由ソフトウェアの大義に改宗させられた新しいユーザー一人あたり、数百、いやおそらく数千人が、まず最初にマイクロソフトの Windows を起動させていた。GNU/Linux は、初歩的なグラフィカル・インターフェースを持っているだけで、それはユーザーに親切なものとは言い難かった。1993 年に、それを使えるのは専門家だけだった。グラフィカル・デスクトップを開発する GNU プロジェクトの最初の試みは失敗していた。

そして政治状況があった。ユーザー・インターフェースの著作権問題は、いまだに現実的脅威だった。まだ法廷は、その観念を拒絶していなかった。一方、アルゴリズムと機能のソフトウェア特許が、他の国々にも拡大しそうな増大する危険になっていた。

最後に、Linux そのものに興味深い性質があった。（GNU のように）設計上のバグや（BSD のように）法的紛争に縛られていなかったが、Linux の急速な進化は非常に無計画で、成功も非常に予想外だったので、このソフトウェアのコードの一番近くにいたプログラ

マたちでさえ、それをどう考えれば良い か分からなかった。オペレーティング・システムというよりも、それはコンピレーション・アルバムで、ハッカーのグレイテスト・ヒット・メドレーからできて いた。つまり、GCC、GDB、glibc(GNU プロジェクトが新たに開発した C ライブラリ)から、X(MIT のコンピュータ科学研究所で開発された Unix ベースのグラフィカル・ユーザ・インタフェース)や、BSD で開発された BIND(Berkeley Internet Naming Daemon、これによりユーザは数値表現の IP アドレスを覚えやすいインターネットドメイン名で置き換えられる)に TCP/IP まで、全部入っていた。加えて、それに Linux カーネルが入っていたが、それ自身は Minix を置き換えるものとして構想されたものだった。自分たちのオペレーティング・システムをゼロから作り上げるのではなく、トーバルズとその急速に拡大する Linux 開発チームは、この基盤に彼らの作品を差し込んだ。トーバルズ自身は、後に彼の成功の秘訣を語ったときに、これをこう言い換えた。「ぼくは元来怠け者で、 本当は他人がやったことを自分の手柄にするのが好きなんだよ。」1

この怠惰さは、効率性の観点からは賞賛されるべきだが、政治的な観点からはやっかいなことだった。たとえば、それはトーバルズにイデオロギー的な計 画性がないことを明確に示していた。GNU の開発者たちとは違い、トーバルズは、仲間のハッカーたちに自由を与えたくて、オペレーティング・システムを 作っていたわけではなかった。彼は、自分自身が遊べる何かが欲しくて作ったのだ。そうすると、正確には、この結合的システムは何なのか、どの哲学で人々は 仲間になっているのか。 ストールマンが GNU 宣言で初めて表明した自由ソフトウェアの哲学の現れだったのか？それとも、同じような動機をもったユーザーなら、自宅のシステム上で組み立てられる気の利 いたソフトウェア・ツールの混合物にすぎなかったのか？

1993 年の暮れまでに、Linux ユーザーのますます多くが、後者の定義に寄りかかり始め、Linux を主題に私的な変種を醸成し始めていた。彼らは、様々な GNU/Linux つまり「ディストリビューション」を開発し、ある時は無料で、ある時は価格を付けて、配布し始めた。その結果には、良く言っても、むらがあった。

「これは、レッドハットや他の商用ディストリビューション以前のことだね」と、当時、パデュー大学のコンピュータ科学の学生だったイアン・マードッ ク（Ian Murdock）は回想する。「Unix の雑誌をパラパラめくるだけで、『Linux』を賛美する名刺大の広告が山ほど見つかるようなありさまさ。ほとんどの会社は、自分たちの[占有的な]ソー ス・コードをほんの少し混ぜ物にすべりこませても何も悪くないという、あやしい作戦をしていたんだ。」

Unix プログラマだったマードックは、Linux を自宅のパソコンのシステムに初めてダウンロードしてインストールしたとき、「引きさらわれた」ことを覚えている。「とにかく楽しかったんだ」と彼は言 う。「それで関わりたくなった。」だが、できの悪いディストリビューションの氾濫が、彼の初めの熱意をそぐようになった。最良の関わり方は無添加版の Linux を作ることだと決めて、マードックは、自分のディストリビューションに取り込むべく、利用可能な最良の自由ソフトウェアツールのリストを作り始めた。「ぼ くは、Linux

の名に恥じない何かがほしかったんだ」と、マードックは言う。

「興味をかきたてよう」として、マードックは自分の考えを、ユーズネットのcomp.os.linux ニュースグループなど、インターネット上に投稿した。最初の返信メールの一つは、rms@ai.mit.edu からのものだった。ハッカーであるマードックは、即座にそのアドレスが誰のものかを理解した。それは GNU プロジェクトの創設者リチャード・M・ストールマンであり、マードックが以前から「ハッカーのなかのハッカー」として知っていた人物だった。メールの一覧 にそのアドレスを見て、マードックは戸惑った。いったいなぜストールマン、つまり自分のオペレーティング・システムのプロジェクトを率いている人物が、 マードックの "Linux" ディストリビューションに対する不満を気にかけるのだろう？

マードックはメールを開いた。

「彼が言うには、自由ソフトウェア財団（FSF）は Linux を詳しく観察し始めていて、FSF は、ことによると Linux [原文のまま]システムでやってみることにも興味を持っている。ストールマンは、ぼくたちのゴールと彼らの哲学が、大筋で一致していると見ているようだっ た。」

誇張ではなく、そのメールは、ストールマン側の戦略の変化を示していた。1993 年まで、ストールマンは Linux で起きていることには、首を突っ込まないことでよしとしていた。新しいカーネルについて最初の知らせを受けたとき、ストールマンは友人に適合性のチェック を託した。ストールマンは思い出して言う、「彼の報告によると、そのソフトウェアは System V をモデルにしていました。System V は、Unix の劣化版でした。それに彼が言うには、それはポータブルじゃなかったんです。」

その友人の報告は正しかった。386 マシン上で動作するように作られていたので、Linux はその低価格なハードウェア・プラットフォームにしっかり根づいていた。だが、友人が報告しそこねたのは、市場で唯一、自由に改変できるオペレーティン グ・システムとして、Linux にかなり大きな利点があることだった。言い換えれば、ストールマンが次の 1 年半を、HURD 開発者からの進捗報告、かなり遅い進歩の報告に耳を傾けて過ごしている間に、トーバルズは、後に Linux と GNU を根こそぎ新しいプラットフォームに移植することになるプログラマたちを味方に引き入れたのだった。

1993 年までに GNU プロジェクトが動作するカーネルを配布できなかったことは、GNU プロジェクトと自由ソフトウェア運動全体に色々な問題を引き起こした。1993 年 3 月、雑誌ワイアードのシムソン・ガーフィンクルによる記事は、GNU プロジェクトを、多くのツールで成功したが、「泥沼にはまってしまった」と形容した。2 プロジェクトとその非営利随伴者である自由ソフトウェア財団内部の人々は、ガーフィンクルの記事が口にしたよりも、ムードはずっと悪かったことを覚えてい る。「そこに新しいオペレーティング・システムを導入するチャンスの窓が開いていることは、少なくとも、ぼくにはそのとき全く明らかでした」と、シャッセ ルは言う。「そして、一度その窓が閉じると、ひとびとは、ほ

とんど興味をなくしてしまうでしょう。実際、まさにそうなってしまいました。」3

1990 年から 1993 年にかけての GNU プロジェクトの苦闘については、多くのことが言われてきた。こうした苦闘をストールマンのせいにする者もいるが、初期の GNU Emacs チームメンバーで、後にストールマンの批判者になるエリック・レイモンドは、ほとんどは組織の問題だ、と言う。「FSF は傲慢になった、」とレイモンドは言う。「目標を、製品として提供可能なオペレーティング・システムを作ることから、オペレーティング・システムを研究す ることに後退させたんだ。」さらに悪いことには、「彼らは、FSF の外の何かから自分たちが影響をこうむるなんて、ありっこないと思ってたんだ。」

マードックは、より寛大な見方をする。「問題の一部は、彼らはちょっと野心的すぎて、見込みのないことにお金を投入しすぎたことだと思う。」と、彼 は言う。「80 年代後半から 90 年代前半まで、マイクロカーネルは、ホットな話題だった。不幸にも、それは GNU プロジェクトがカーネルの設計をはじめた頃のことだ。彼らは多くのお荷物を抱え、それを捨て去るのに多くの後戻りを必要としたんだ。」

ストールマンの反応は次のようなものだ。「レイモンドの言う感情は彼の想像力から来ていますが、Hurd の遅れの原因について彼は正しいことを一つ言っています。Hurd の開発者は、すぐに Hurd が走るようにするよりも、彼が学んだことに基づいてコードの大部分を再設計し書き直しました。それは良い設計の実践でしたが、動作するものを出来るだけ早く作るという我々の目標からすれば、正しい実践ではありませんでした。」

遅れを説明して、ストールマンは多くの問題を挙げている。ロータスとアップルの訴訟にさらに注意を払う必要があった。これに、ストールマンが三年間 タイプできなくなくなった問題が重なった。その間、ストールマンはほとんどプログラミングができなかった。また、ストールマンは GNU プロジェクトの色々な部門間のコミュニケーションの悪さを挙げる。「デバッグ環境が機能するように、多くの仕事をする必要がありました。」と彼は回想す る。「当時、GDB のメンテナーたちは、協力的じゃなかったんです。」彼らは、GNU システムの完成という全般的な目標よりも、既存のプラットフォームの、GDB の現在のユーザーを優先していたのだった。

でも最大の原因は、とストールマンは言う、彼や GNU プロジェクト・チームの他のメンバーが、Mach マイクロカーネルを、本格的な Unix カーネルに拡張することの難しさを過小評価していたことだった。「大丈夫、マシンとやりとりをする［Mach の］部分はデバッグ済だ、とぼくは思ったんで す。」とストールマンは、2000 年のスピーチで HURD チームのトラブルを回想しながら、言う。「有利なスタートが切れているから、早く終れるに違いない。しかし、それどころか、こういう非同期なマルチスレッ ドのプログラムをデバッグするのは、本当に大変だということが分かったんですね。タイミングバグがあって、それがファイルを上書きしてしまう。楽しむどこ ろじゃない。結局、テストバージョンを作るのに何年もかかってしまいました。」4

そのうち、GNU と Linux の組み合わせがますます成功していることが、GNU プロジ

ェクトは HURD を待つべきじゃなくて、すでに出発した列車に乗るべきだということを明白にした。その上、GNU/Linux を取り巻くコミュニティには弱点があった。確かに、Linux は GPL でライセンスされていたが、マードック自身が注意していたように、GNU/Linux を純粋に自由なオペレーティング・システムとして扱う欲求は全員一致からは遠かった。1993 年末に、Linux ユーザーの総人口は、1 ダースかそこらの Minix ファンから、2 万人から 10 万人の間のどこかにまで増大していた。5 かつて趣味だったものが、今や、果実をもぎ取ることのできる市場になっていた。そして、開発者の中にはそれを自由ではないソフトウェアをくっつけて収穫す ることに異論のない者もいたのだった。ベルリンに向かって辺りを払って進んでいくソビエト軍を見たウィンストン・チャーチルのように、Linux の「勝利」を祝う時の到来に、ストールマンが複雑な感慨に襲われたことは理解できる。6

パーティーに遅れてやってきたとはいえ、ストールマンはまだ強い影響力があった。FSF がマードックのソフトウェア・プロジェクトを資金的・精神的に支援することを公表した直後から、他の支援の申し出が舞い込み始めた。マードックはこの新規 プロジェクトを、Debian（デビアン） -- 彼と彼の妻デボラの名前をくっつけて縮めたもの -- と名づけ、二、三週間後には最初の配布物がお目見えした。「[リチャードの支援が] Debian を一夜にして、面白そうな小プロジェクトからコミュニティーの人々が注目せずにはいられないプロジェクトに一変させた」とマードックは言う。

1994 年 1 月、マードックは「Debian 宣言」を起草した。10 年前のストールマンの「GNU 宣言」の精神にならって書かれた宣言は、自由ソフトウェア財団と緊密に連携することの意義を説明した。マードックはこう書いている。

これからの Debian にとって、自由ソフトウェア財団が果たす役割は極めて重要だ。彼らがそれを配布するという単純な事実により、メッセージが世界に送られるのである。それは、Linux は商業的産物ではなく全然そうあるべきでもないが、そのことが Linux に商業的競争力がないことを意味しないというメッセージである。これに同意できない人は、GNU Emacs や GCC が成功した理由を説明してほしい。これらは、商用ソフトウェアではないという事実にもかかわらず、商業市場に極めて大きな衝撃を与えたのである。 Linux コミュニティー全体とその将来を犠牲にして自分が金持ちになるというような破壊的目標ではなく、Linux の将来に集中すべき時が来た。Debian の開発や配布は、この宣言でアウトラインを描いた諸問題に対する回答ではないかもしれないが、それが少なくともこうした諸問題に多くの注意を引きつけるこ とで、それらを解決できるようになることを願っている。7

宣言を発表してまもなく、自由ソフトウェア財団は、最初の大きな要求を出してきた。ストールマンは、マードックがそのディストリビューションを GNU/Linux と呼んで欲しいと思った。最初、ストールマンは -- その心臓部に GNU を持つ Linux という意味で-- Lignux という用語を提案したが、その最初の反応が非常に否定的なものだったので、スト

ールマンはそれよりも批判の小さな GNU/Linux で行くことを納得した。
　ストールマンが頭に GNU をつけさせようとするのを、遅すぎるクレジット要求だとして退ける人もいたが、それが義務的かどうかは気にするまでもない。マードックは違う見方をした。 振り返ってみて、マードックは、それを GNU プロジェクトの開発者と Linux カーネルで一緒に使うために GNU プログラムを変更した人々との間で高まっていた緊張をほぐそうとしてのことだと見る。「亀裂が見え始めていて」とマードックは振り返る。「リチャードはそ れを気にしたんだ。」
　1990 年までに、どの GNU プログラムにも指名された担当者のメンテナーがいた。いくつかの GNU プログラムは、多くの異なるシステムで走らせることができた。そして、しばしばユーザーはそれらの他のシステムへの移植のための変更に貢献した。これらの ユーザーはその一つのシステムを知っているだけのことが多く、他のシステムのためにコードをきれいに維持する方法を考えていなかった。新しいシステムをサ ポートする一方、読みやすいコードを維持することで、全てのシステムで信頼性を維持することができた。それには多くの変更を書き直す必要があった。担当者 のメンテナーは変更に批判を加えて、ユーザーの作者に移植の書き直し部分のやり方を知らせる責任があった。一般に、ユーザーは、変更が標準ヴァージョンに 統合されるようにするために、これに熱心だった。それから、担当のメンテナーはやり直された変更を編集し、それ以降のメンテナンスを引き受けた。いくつか の GNU プログラムにとって、これは、何十個の異なるシステムで何十回も繰り返されてきたことだった。
　様々な GNU プログラムを Linux カーネルとともに動くように適応させたプログラマは、この共通の経路をたどった。彼らは、彼ら自身のプラットフォームだけを考えていた。しかし、将来のメ ンテナンスのために、担当者のメンテナーが彼らの変更をきれいに仕上げるための助力を求めた時、彼らのうちの何人かは興味を示さなかった。彼らは間違いのないようにする事、彼らが手を加えた GNU パッケージの将来のメンテナンスを楽にする事に関心はなかった。彼らは自分たちのヴァージョンのみを気にかけており、それらをフォークとして続行すること に傾いていた。
　ハッカーの世界では、フォークは興味深い問題だ。ハッカー倫理では、プログラマは所与のプログラムのソースコードに対して、したいことを何をしてい もいいが、オリジナルの開発者がメンテナンスしている共通ヴァージョンに成果を提供するのが適切な行動と考えられている。ハッカーは、自分たちの改良をプ ログラムの主要なヴァージョンに注力することが、通常は実用的だし、好ましいと思うのだ。自由ソフトウェアのライセンスは、全てのハッカーにプログラムを フォークする権利を与えていて、時にそれが必要になるが、必要性や理由もなくフォークするのは、少々お粗末な行為と考えられているのである。
　GNU プロジェクトのリーダーとして、ストールマンはすでに 1991 年に、ソフトウェアをフォークさせることのネガティブな影響を経験していた。ストールマンは言う、「ルーシド社 (Lucid) は、GNU Emacs の改良を書くために何人か雇いました。それは GNU Emacs

への貢献を意味していました。しかし、開発者たちはプロジェクトについて知らせてくれなかった。そのかわりに、そのまま幾つかの新機能を設計しました。き みの予想する通り、ぼくは彼らの決定の或るものは同意し、他のものは同意しませんでした。彼らは彼らのコード全てを統合することを求め、ぼくが半分だけ使 いたいと言ったとき、彼らはその半分について適応させる作業を手伝うことをやめました。ぼくは自分でそれをしないといけませんでした。」 このフォークは並行バージョンである Lucid Emacs を生み出し、至るところに気まずい雰囲気が漂った。8

今やプログラマは主な GNU パッケージの幾つかで一挙にフォークしていた。最初、フォークは性急さの産物だと考えたとストールマンは言う。Linux チームの素早く形式ばらない動きと対照的に、GNU のソースコードのメンテナーたちは、よりゆっくりしていて、プログラムの長期的な存続に影響するような変更には、より慎重だった。彼らはまた、他人のコー ドを厳しく批判することも恐れなかった。しかしながら、ストールマンは、Linux 開発者たちのメールを読んでいるうちに、GNU プロジェクトとその目的について認識の根本的な欠落があることに気づき始めた。

「自分を Linux ユーザーだと考えている人たちは、GNU プロジェクトを大切に思っていないことを、ぼくらは発見しました、」とストールマンは言う。「『何でそんなもの気にしなきゃいけないの？ GNU プロジェクトなんて気にしないよ。それはぼくの役に立っているし、ぼくら Linux ユーザーに都合がいいけど、どうでもいいよ。』と彼らは言う。本質的には一種の GNU システムを使っているのにほとんど気にしていない、というのは全く驚きでした。彼らは GNU のことを誰よりも気にかけないのでした。」自分たちの実践を Linux の組み合わせと呼ぶことに騙されて、彼らは彼らのシステムが Linux というよりも GNU であることを実感していなかった。

団結のために、ストールマンは担当のメンテナーに通常は変更する作者がするべき作業をすることを要請した。ほとんどのケースでこれは実現可能だった が、glibc では、そうはいかなかった。glibc とは、GNU C ライブラリの短縮語で、それは全てのプログラムがカーネルに向けて「システムコール」をするために使うパッケージだが、この場合 Linux では、Unix 風システム上のユーザープログラムは、C ライブラリを通してのみカーネルとやりとりしている。

glibc を Linux とシステムの全ての他のプログラムとの間のコミュニケーション経路として働くようにする変更は、広範囲でアドホックなもので、他のプラットフォームへのそれらの影響を考慮することなく書かれたものだった。glibc の担当メンテナーからすれば、それをきちんと書改めるのは気持ちを萎えさささせる仕事だった。自由ソフトウェア財団は彼にスクラッチから再実装するために約 一年間を使わせるかわりに、glibc のヴァージョン 6 を GNU/Linux の箱からそのまま働くものにした。

マードックは、Debian のソフトウェア・ディストリビューションを展開するときに、ストールマンが GNU の接頭辞を加えるように急いで主張する原因となったのはこれだった

と言う。「それ以降、フォークは終息しました。でも、その時は Linux のコミュニティが自分たちを GNU コミュニティと別と見るなら、それが分裂の力になるかも知れなかったんですよ。」

Linux を GNU プロジェクトの「一種」と記述するのを、政治的貪欲と見る人もいるが、以前から自由なソフトウェアの大義に共鳴していたマードックは、ストールマンが、Debian 版を GNU/Linux と呼ぶように求めたのはリーズナブルだと思った。「名誉というよりも、団結を求めたんだ」と彼は言う。

すぐに続けて、より技術的な性格の要求があった。マードックは政治的な問題には合わせていこうとしたが、実際のソフトウェアの設計や開発モデルのことになると、もっと断固とした姿勢をとった。連帯ショーとして始まったことが、間もなく意見の相違の繰り返しになった。

「僕にも彼と一致できないところがあったんですよ、」とマードックは笑いながら言った。「正直言って、リチャードは一緒に仕事するのがとても難しい ことがある人なんだ。」最も重要な不一致はデバッギングについてだった。ストールマンは、ユーザーが遭遇するかもしれない如何なるバグもすぐに調査できる ように、デバッギング情報を全ての実行可能なプログラムに含めることを望んだ。マードッグは、これはシステムファイルを大きくしすぎて配布の邪魔になるだ ろうと考えた。どちらも考えを変えたがらなかった。

1996 年、マードックはパデュー大学を卒業すると、成長する Debian プロジェクトの手綱を手渡そうと決めた。既に彼は、管理の仕事をブルース・ペレンズに引き継いでいた。ペレンズは GPL でリリースされている Unix ユーティリティの Electric Fence の作者として有名なハッカーだ。ペレンズは、マードック同様、そのオペレーティング・システムの Unix 風の能力が明らかになるとすぐに、GNU/Linux に魅了された。マードックと同様、ペレンズはストールマンと自由ソフトウェア財団の政治的課題に、遠くからではあるが、共鳴していた。

「ストールマンが既に GNU 宣言や GNU Emacs や、GCC を携えて登場した後で、彼がインテルのコンサルタントとして働いている、という記事を読んだのを思い出します」と、1980 年代末の最初のストールマンとの接触を思い出して、ペレンズは言う。「彼に、どうすれば一方で自由なソフトウェアを提唱して、他方ではインテルのために働 けるのかと尋ねる手紙を書いたんです。彼は返事をくれて『僕は自由なソフトウェアをつくるためにコンサルタントとして働いています。』と言うんですね。彼 はとても丁寧で、その答えは全く道理にかなっていると思いました。」

しかし、著明な Debian 開発者の一人として、ペレンズは、マードックが設計のことでストールマンと争っていたことは残念だった。開発チームのリーダーになってすぐ、Debian は自由ソフトウェア財団と距離を置くことを指示する決定をしたとペレンズは言う。「ぼくらは、リチャードの細かい管理スタイルを望まないと決めました。」

ペレンズによれば、ストールマンはこの決定に不意をうたれたが、それを認める知恵を持

っていた。「一定の冷却期間をおいて、彼は我々が真に関係を結 ぶ必要があるとメッセージを送ってきました。彼は、GNU/Linux と呼ぶことを注文したが、それだけでした。これは素晴らしいと確信しました。僕は一方的にそう決めました。みんな、ほっとしました。」

時とともに、Debian は、Slackware（スラックウェア）と並ぶ、ハッカー版 GNU/Linux としての声価を高めていくことになる。Slackware は、Debian と同じく 1993 年から 1994 年にかけて立ち上げられたもう一つの人気ディストリビューションである。しかし、Slackware には自由ではないプログラムが幾つか含まれていて、Debian も、GNU との分離後に自由ではないプログラムを配布し始めた。9「自由ではない」とラベルをつけて「Debian の公式的な部分ではない」と言うものの、ユーザーに提供するのは、それとなく裏書きしているようなものだった。GNU プロジェクトがこれらのポリシーに気 付いたとき、Slackware も Debian も、公衆に推薦できる GNU/Linux のディストロではないことが認識されたのだった。

だが、ハッカー指向のシステムの世界の外で、GNU/Linux は商用 Unix 市場から力を集め始めていた。ノースカロライナでは、レッドハットという名前で売り出していた Unix の会社がビジネスを GNU/Linux へ方向転換しようとしていた。代表取締役の ロバート・ヤングは、かつての Linux ジャーナルの編集者で、1994 年に、リーナス・トーバルズへ質問して、カーネルを GPL の下に置いたことを後悔していないか尋ねたことがある。ヤングによれば、トーバルズの返事は、彼の GNU/Linux 観に「深遠な」衝撃を与えた。ソフトウェアの昔ながらの戦術で GNU/Linux を買い取る方法を探るのではなく、ヤングは Debian と同じアプローチつまり完全に自由なソフトウェア部品だけでできたオペレーティング・システムをつくるというアプローチをとる会社が現れたら何が起こるか について考え始めた。シグナス・ソリューションズは、1990 年にマイケル・ティーマンとジョン・ギルモアが立ち上げた会社だが、既に品質と特注生産能力を基盤に自由ソフトウェアが売れることを実証していた。レッド ハットが、GNU/Linux について同じアプローチをとったとしたらどうなるだろうか？

「西洋の科学的伝統の中で、我々は巨人の肩に乗っているのです、」とヤングは、彼以前のトーバルズやサー・アイザック・ニュートンに倣って言う。 「これをビジネスの言葉で言うと、仕事を進める上で車輪を二度発明しなくていい、ということです。[GPL の]モデルの美しいところは、自分のコードをパ ブリック・ドメインにするところだ。9 例えば、君が独立のソフトウェアベンダーで、あるアプリケーションを造ろうとしていて、モデム・ダイアラーが必要なとき、何でモデム・ダイアラーを二度発 明するんでしょうか？君は、Red Hat Linux から PPP を拝借して、君のモデム・ダイアルツールの核として使うことができる。もしも、グラフィックツールが一セット必要なら、自分のグラフィックライブラリを書 く必要はない。GTK をダウンロードするだけでいい。一瞬にして、既にあるものから最良のものを再利用できるようになります。そして、その瞬間から、アプリケーションベンダー である君は、ソフトウェア・マネージメントより、顧客のニーズに特化したアプリケ

ーションを書くことに多くを注力するようになるのです。」しかしながら、 ヤングは自由ソフトウェアの活動家ではなく、喜んで自由ではないプログラムをレッドハットのGNU/Linux システムに含めている。

自由ソフトウェアのビジネス上の効率性に関心を示したソフトウェア経営者はヤングだけではなかった。1996 年末までに、ほとんどの Unix の会社は目を覚まし、熟成したソースコードの匂いに鼻を動めかし始めた。Linux 部門が商業的に完全にブレークするのは、まだ一、二年先のことだが、ハッカー・コミュニティの十分近くにいた者は感じることができた。何か大きなことが起 こりつつあることを。インテルの 386 チップ、インターネット、そしてウェブのセットが、大波のように市場に打ち寄せていた。そして、自由ソフトウェアは、かつてないほどの大波に見えた。

イアン・マードックにとって、この波は、自由ソフトウェア運動にアイデンティティを与えることに多くの時間を費やしてきた男への感謝の贈り物である と同時に罰であるように思えた。多くの Linux ファンの例にもれず、マードックは Linux が最初にポストされたところを目撃していた。Linux は「ただの趣味」というトーバルズの最初の警告を見た。また、Minix をつくったアンドリュー・タネンバウムに対するトーバルズの「去年の春にGNU カーネルが存在していたら、僕はこのプロジェクトを開始しなかっただろう」という告白を見た。11 多くの人たちと同じく、マードックは、何度も機会が浪費されたことを知っていた。彼はまた、インターネットの織物から新たな機会が浸み出てくるのを目撃す ることの興奮も知った。

「あの早い時期に Linux とかかわっていたのは楽しかったですね、」とマードックは回想する。「それはすべきことがあるということであり、同時に、時間をつぶせるものがあるという ことでした。昔の [comp.os.minix] ニュースグループでのやりとりを読み返してみると、その気持ちがわかるでしょう。これは、HURD ができるまで、僕らが遊べる何かだということでした。みんな気を揉んでいたんですよ。面白いもんですね。色んな意味で、HURD がもう少し早くできていたら、Linux は絶対現れなかったんじゃないかな。」

だが、1996 年末までに、こうした「だとしたら」の問いかけは、すでに無意味なものになった。ユーザーたちが語っていたのは、それを Linux と呼ぶか、GNU/Linux と呼ぶか、ということだった。36 ヵ月の窓は閉じられた。それは、GNU プロジェクトが HURD カーネルを公開しても、それに注目しているハードコアなハッカーコミュニティを除けば、チャンスはほとんどない、ということだった。GNU システムの最後のギャップを埋めることで、Linux は、Unix 風の自由ソフトウェアのオペレーティングシステムをつくる GNU プロジェクトの目標を達成した。しかし、大半のユーザーは何が起きたのかを認識していなかった。彼らはシステム全体を Linux と考え、リーナス・トーバルズがそのすべてを成し遂げたと考えた。彼らのほとんどは、自由ではないソフトウェアを同梱し、トーバルズを倫理的ガイドにした ディストリビューションをインストールした。彼らには、それを拒絶する道徳的理由は見えなかった。それでも、心もとない自由は、それを賞賛する人には利用 可

能になった。

後注

1 ここで引用したような発言を、トーバルズは複数の機会に行っている。だが、今のところ、この引用で一番有名なのは、エリック・レイモンドのエッセー『伽藍とバザール』(1997年5月)の中のものだ。http://catb.org/esr/writings/homesteading/cathedral-bazaar/(日本語訳: http://cruel.org/freeware/cathedral.html)

2 シムソン・ガーフィンケル『ストールマン失速か？』Wired(1993年3月)を参照。

3 新しいオペレーティング・システムに36ヵ月間だけ開いていた「窓」に関するシャッセルの懸念は、GNUプロジェクトに固有の問題ではない。1990年代の始め、自由ソフトウェア版のBSDは、BSDからの派生ソフトウェアのリリースを制限しようとするUnixシステム研究所の訴訟によって、お手上げ状態になった。多くのユーザーが、BSDから分家したFreeBSDやOpenBSDは、性能面でもセキュリティ面でもGNU/Linuxを凌駕していると思っているが、FreeBSDやOpenBSDのユーザー数は、GNU/Linuxユーザーの総人口に比べれば、ほんの一握りに過ぎない。

GNU/Linuxが、他の自由ソフトウェアのオペレーティング・システムに対して、相対的に成功したことの分析例としては、ニュージーランドのハッカー、リアム・グリーン ウッド(Liam Greenwood)のエッセー『Linux成功の理由』(1999年)を参照。http://www.freebsddiary.org/linux.php

4 マウイ高性能計算センターでのスピーチを参照。

後日、私はメールで、「タイミングバグ」の正確な意味を尋いてみた。ストールマンによれば、「タイミングエラー」と言う方が問題を適切に要約してい るとのことで、タイミングエラーがオペレーティング・システムの実行をどのように妨げるかについて、明解な技術情報を提供してくれた。

「タイミングエラー」は非同期システムの中で発生します。非同期システムでは、並行 して実行されるジョブは、理論的には任意の順序で起動可能ですが、ある特定の順序で起動されたときに問題が発生します。プログラムAが行う処理をX、プログラムBが行う処理をYとし、XもYも同一のデータ構造を検査して更新する短い処理だとしましょう。ほとんどの場合、コンピュータは、Yの前にXを行うか、Xの前にYを行うかで問題は起こりません。だが、稀に、たまたまスケジューラがプログラムAを実行し、処理Xの途中で、処理Yを行うプログラムBを実行する場合があります。すると、YはXの途中で実行されることになる。これらは同じデータ構造を更新するので、互いに影響し合うのです。例えば、Xはおそらくデータ構造の検査を終っていて、変更があったことを気づかないでしょう。そこで、再現しづらいエラーが発生することになる。再現しづらいのは、 偶然の要因に(スケジューラがどのプログラムをどれだけの時間実行するかを決定するときのような)左右されるからです。こうしたエラーを回避するには、XとYが同時に実行されないことを保証するために、ロックを用いる方法があります。非同期システムを書くプログ

ラマーは、ロックの一般的な必要性は理解している のですが、ときどき、特定の場所やデータ構造で必要なロックを見過ごします。すると、そのプログラムにはタイミングエラーが存在することになるのです。

5 GNU/Linux のユーザ人口は、せいぜい概算しか手に入らないため、こんなに範囲が広くなっている。10 万人という数字は、レッドハットの『マイルストーン』からのもの。 http://www.redhat.com/about/corporate/milestones.html

6. このようにウィンストン・チャーチルになぞらえたのは、ストールマン自身がチャーチルに関する次のコメントを、頼まれないのに送ってくれたときよりも前のことである。

第二次世界大戦とその勝利に必要とされた決意とは、ぼくの成長期の強烈な思い出だ。「我々は上陸地点で彼らと戦うのだ、浜辺で彼らと戦うのだ ... 我々は絶対に降伏しない」というようなチャーチルの演説は、いつもぼくの中で鳴り響いていた。

7 イアン・マードック『Debian 小史』(1994 年 1 月 6 日) : 付録 A、『Debian 宣言』。 http://www.debian.org/doc/manuals/project-history/apA.html

8. 元 Lucid のプログラマーで、後に Mozilla 開発チームを率いることになる、ジェイミー・ザウィンスキー（Jamie Zawinski）のウェブサイトに、Lucid/GNU Emacs のフォークに関するドキュメントがある。タイトルは、『Lemacs/FSFmacs の分裂』。 http://www.jwz.org/doc/lemacs.html

9 1996 年 6 月の Debian Buzz は、その Contrib section に自由なものではない Netscape 3.01 を含んでいた。

10 ヤングはここで「パブリックドメイン」という用語をおおざっぱに使っている。厳密には、パブリックドメインは著作権が適用されないものである。GPL でリリースされたプログラムは、パブリックドメインではありえない。なぜなら、GPL を適用するためにそれは著作権が適用されているからだ。

11 これは、Linux の最初のリリースに続いて起こった、トーバルズ-タネンバウム間の大いに宣伝された「炎上」からの引用だ。移植性がなく一枚岩的なカーネルの設計を選択し たことを弁護しようとして、トーバルズは、Linux の開発を始めたのは、新しい 386 PC をもっとよく知るためだったと言う。「GNU のカーネルが去年の春に出来上がっていたら、ぼくは、ぼくのプロジェクトを始めようとすらしなかっただろう。」クリス・ディボナ他『オープンソース』 （O'Reilly & Associates, Inc.1999 年）224 ページを参照。

第十一章　オープンソース

[RMS: この章でのみ引用文をいくつか削除した。削除した内容はオープンソースに関するもので、私の人生または仕事と無関係なものだった。]

1995 年 11 月、自由ソフトウェア財団のメンバーで 1994 年に出版された「UNIX の 1/4 世紀」という本の著者でもあるピーター・サルス（Peter Salus）は、GNU プロジェクトの「システムに関する議論用」メーリングリストのメンバーに論文を募集した。カンファレンスの議長に予定されていたサルスは、まもなくマサ チューセッツ州ケンブリッジで開催される、自由に再配布可能なソフトウェアをテーマにしたカンファレンスを、仲間のハッカーに知らせておきたかったのだ。 1996 年 2 月に予定され、自由ソフトウェア財団がスポンサーをするそのイベントは、自由ソフトウェアだけを対象とした初の技術会議だったが、GNU 以外の自由ソフトウェアのプログラマとの団結を示すために「GNU、Linux、NetBSD、386BSD、FreeBSD、Perl、Tcl/tk、 その他の、コードにアクセスすることができて再配布可能なツール」に関する論文を歓迎した。サルスは書いている。

この 15 年間に、自由で低コストなソフトウェアが至るところに存在するようになりま した。このカンファレンスは、いろいろ異なるタイプの、自由に再配布可能なソフトウェアの実装者とそうしたソフトウェアの（色々なメディアでの）パブリッ シャーを引き合わせることになるでしょう。チュートリアルや参照論文が用意され、リーナス・トーバルズとリチャード・ストールマンが基調講演を行います。 1

サルスのメールを受け取った人の中にカンファレンスの委員のエリック・S・レイモンドがいた。そのメーリングリストの他のメンバーのようにプロジェ クや会社のリーダーではなかったものの、いくつかのソフトウェア・プロジェクトと、ガイ・スチール（Guy Steele）が十年前に出版したハッカー辞典の大増補版の新ハッカー辞典の編者をしたことにより、レイモンドは、ハッカーコミュニティの中でかなりの評 価を得ていた。

レイモンドにとって、1996 年のカンファレンスは願ってもないイベントだった。彼は自由ソフトウェア運動の理想を完全に支持していたわけではなかったが、いくつかの GNU プログラム、とくに GNU Emacs に貢献していた。それらの貢献は 1992 年、レイモンドがオフィシャルな GNU ヴァージョンの GNU Emacs をストールマンと議論せずに変更する権限を要求したときに終わった。ストールマンはその要求を断り、レイモンドはストールマンの「細かいことに口を出す運 営」を非難した。「僕が Emacs LISP ライブラリをきれいに書き換えようとしたら、リチャードは僕が無権限で修正することに大騒ぎしたんだ」とレイモンドは回想する。「とても失望して、彼とは もう一緒に作業しないと決めたんだ。」

仲たがいはあったが、レイモンドは自由ソフトウェアのコミュニティで活動を続けた。それで、サルスがカンファレンスでストールマンとトーバルズの二 人に基調講演をさせることを提案したとき、レイモンドは逸る思いでその考えを支持した。ストールマンが年長で知

恵豊かな ITS/Unix ハッカーを代表し、トーバルズが若くてエネルギッシュな Linux ハッカーの一団を代表することで、この組合わせは団結を示す象徴的なショーであり、レイモンドのような野心的で若めの（つまり、40 歳以下の）ハッカーにだけ特に得るところが大きかった。「僕は両方の陣営に足をつっこんでいたからね」とレイモンドは言う。

カンファレンスまでに、二つの陣営の間の緊張は明白になった。それでも両陣営に共通していることが一つあった。カンファレンスはフィンランドから来 た生身の神童に会う初めての機会だったのだ。意外にも、トーバルズは魅力的で人好きのする講演者だった。わずかにスウェーデン訛りがあったが、トーバルズ は、素早い控え目なウィットで聴衆を驚かせた。2

もっと驚いたのは、とレイモンドは言う、最も著名なハッカーのリチャード・ストールマンを含む、他の著名なハッカーに対してトーバルズが等しく言い たい放題だったことだ。カンファレンスの終わるときまでに、トーバルズの半ばハッカー的で半ばスラッカー的な態度は、カンファレンス参加者の年長組も若手 も同じように惹きつけていた。

「あれが転機だった」とレイモンドは回想する。「1996 年以前は、リチャードがハッカー文化全体のイデオロギー上の指導者の資格を満たす唯一の人間だった。彼と意見を異にしても、公然と異を唱える者はいなかった。そのタブーを打ち破ったのがトーバルズだ。」

究極のタブー破りは、このショーの終り近くにやってきた。増大するマイクロソフトの Windows の市場支配やそれに類する話題を議論しているときに、トーバルズはマイクロソフトのパワーポイントというスライドショー・ソフトウェアプログラムのファン であることを認めた。伝統的なソフトウェア純粋主義の立場からすれば、それは奴隷廃止論の集会で自分の奴隷の自慢をするようなものだった。トーバルズや増 えつつある彼のフォロワー達の立場からすれば、それはただの常識だった。どうして主義主張だけで便利な占有的ソフトウェアプログラムを遠ざけるんだ？彼ら はその点について全く同意しなかった。自由が犠牲を求めたとき、自由を気にしない人々にとっては、それは禁欲に見えて、何か重要なものを獲得する方法には 見えなかった。ハッカーであることは、禁欲ではなく、仕事を成し遂げることだ。そして「仕事（the job）」が彼らの実用語における定義だった。

「あれを言うのはかなりショッキングだった」とレイモンドは振りかえる。「しかし、1995 年と 1996 年までに彼は急激に影響力を獲得しつつあったから、それができたんだ。」

ストールマンの方では、1996 年のカンファレンスで緊張があった覚えはない。たぶん、トーバルズがそれを発言したときにいなかったのだ。しかし、後にトーバルズの有名な「生意気さ」の 棘を感じたのを覚えている。「Linux の文書に、GNU のコーディングの標準をプリントアウトして破り捨てろ、と言っているものがありました」とストールマンはその一例を語る。「良く見ると、彼が不同意だとい うのは、それの最も重要性の小さい部分で、推奨する C コードの書き方の部分でした。」

「よし分かった、つまり彼は我々の習慣のいくつかに異論があるんだ。いいんじゃないですか。でも、それを言うのに随分不愉快なやり方を選ぶんです ね。『こういう方法でコード

をインデントするべきだ』とだけ言えばよかったのに。それでいいんじゃないですか。そこに敵意はないはずです。」

レイモンドにとって、他のハッカーがトーバルズのコメントに寄せた暖かい歓迎は、彼の疑念を裏付けていた。すなわち、Linux 開発者と GNU 開発者を隔てている境界線は、ほとんど世代的なものだった。トーバルズのような多くの Linux ハッカーは、占有的ソフトウェアの世界の中で育ってきた。彼らは自由でないソフトウェアに何か不正を見て自由ソフトウェアに貢献し始めたのではなかった。 彼らの大半にとって、便利さを超える問題はなかった。技術的にプログラムが劣っていなければ、ライセンスの問題だけでプログラムを拒む理由はなかった。い つかハッカーがパワーポイントに代わる自由ソフトウェアを開発するかもしれない。そうなるまで、なぜマイクロソフトやパワーポイントを批判するのか、使え ばいいのに。

これは自由そのものの価値を認める人々と、主にソフトウェアの強力さと信頼性に価値を置く人々との間の、自由ソフトウェア・コミュニティ内の不和の 一例だった。ストールマンは、前者を「自由党」と呼んで、コミュニティ内の政党のように二つの陣営に言及した。他方の陣営の支持者はそれに名前をつけよう としなかったので、ストールマンは軽蔑的にそれを「バンドワゴン党」とか「成功党」と呼んだ。彼らの多くが第一目標として「より多くのユーザーの獲得」を 掲げていたからだった。

GNU プロジェクトを開始して以来の十年間で、ストールマンはプログラマとして、とてつもない名声を築いた。そしてソフトウェアの設計のマネージメントと人のマ ネージメントとの両方で非妥協的だという評判も築いた。これは部分的には真実だったが、評判はまた、ストールマンがしてもいないことを、誰でも好きなとき に引き合いに出す便利な口実を提供した。評判は誤った推測を増幅させた。たとえば、1996 年のカンファレンスの少し前に、自由ソフトウェア財団は職員が完全に欠員状態になるという経験をした。それらの職員の辞職後にサルスが雇った自由ソフト ウェア財団の現在の職員、ブライアン・ニューマンス（Brian Youmans）は、当時の様子を回想する。「一時、ピーター[・サルス]だけがオフィスで働いている職員だったんです。」ブリット・ブラッドリー （Bryt Bradley）が彼女の友人に 1995 年に語ったように、前の職員たちは事務局長のせいで不幸だった。

[名称省略]（FSF の事務局長）は、先週、医療的/政治的な休暇から戻ってくる決定をしました。オフィスのスタッフ（ゲナ・ビーン、マイク・ドレイン、そして私）は、彼女が 休職前にした多くの職務上の誤りからすると、彼女が上司では働いていけないと確信しました。さらに、個人が不当な解雇の脅しを受けたという多くの事実、オ フィスのスタッフすべてに対する暴言と言うべき多くの事実がありました。私たちは（何度も）彼女に、上司としては復帰しないように求め、しかし同僚として ならば喜んで働こうと言いました。私たちの求めは無視されました。私たちは辞めました。

問題の事務局長は、次に、ストールマンに最後通牒を与えた。オフィスの全ての権限を彼女に与えなければ辞めるというのである。FSF の理事長として、ストールマンは、彼女に

活動の全権を与えることを断った。それで彼女は辞職し、彼女の替わりに、ストールマンはピーター・サルスを招集し たのだった。

これらの人々が FSF を去ったと聞いたとき、部外者であるレイモンドは、ストールマンのせいだと思い込んだ。これは、GNU プロジェクトのすべての問題はストールマンのパーソナリティにあるとする彼の説の裏付けを提供するものだった。

レイモンドにはもう一つ持論があった。当時の HURD の開発遅延や Lucid-Emacs の分裂のようなトラブルは、通常、ソフトウェアコードの開発ではなく、ソフトウェアプロジェクトの運営に関係するものだ、というのである。

自由に再配布可能なソフトウェアのカンファレンスが終わると、レイモンドはすぐに自分のお気に入りのソフトウェア・プロジェクトに取り組み始めた。 「フェッチメール（fetchmail)」という名前のメール受信ユーティリティである。トーバルズがきっかけになって、レイモンドは、ソースコードをで きるだけ早く頻繁に更新する約束をつけた自分のプログラムを公開した。ユーザーがバグレポートや機能提案を送ってくるようになれば、ごちゃごちゃになると 最初レイモンドは予想したが、できあがったソフトウェアは驚くほど堅牢になるのがわかった。トーバルズ的なアプローチの成功を分析して、レイモンドはすぐに分析結果を発表した。インターネットを「ペトリ皿」にして、そしてハッカーコミュニティの厳しい精査を自然淘汰の一形式として使うことで、トーバルズは 集権的計画ではない進化モデルを創造したのだ。

その上、とレイモンドは断ずる、トーバルズはブルックスの法則をすり抜ける方法を見つけた。IBM の OS/360 プロジェクトのマネージャで、1975 年に出版された『人月の神話』の著者、フレッド・P・ブルックス（Fred P. Brooks）が初めて明確に語ったブルックスの法則は、プロジェクトに開発者を増員してもプロジェクトを更に遅らせる結果にしかならない、と考える。大 部分のハッカーと同様に、ソフトウェアはスープのように限られた人数のコックで作るのが良いと信じていたレイモンドは、何か革命的なものが作用していると 感じた。調理場にコックをどんどん招くことで、実は、トーバルズは出来上がるソフトウェアをより良いものにする方法を見つけていたのだ。3

レイモンドは自分の観察を紙に書き出した。彼はそれをスピーチに仕立てると、ペンシルベニア州チェスター郡の友人と隣人のグループの前で早速披露し た。「伽藍とバザール（The Cathedral and the Bazaar)」と名付けたそのスピーチは、GNU プロジェクトの運営スタイルと、トーバルズやカーネル・ハッカーたちの運営スタイルとを対比していた。

反応は熱狂的だったとレイモンドは言う、だがドイツの GNU/Linux ユーザーが集まった翌年春の 1997 年 Linux 会議で受けた熱狂にはとうてい及ばなかった。

「その会議では講演の終りにスタンディングオベイションを受けたんだ、」とレイモンドは振りかえる。「僕はそれは二つの意味で重要だと思った。一つ には、彼らが講演内容に興奮したということ。もう一つは、言葉の壁のある講演を聞き終わった後なのに彼らが興奮していたことだ。」

最終的にレイモンドはその講演を、同じ『伽藍とバザール』と題した論文に書き直す。その論文の題名は、レイモンドによる議論の中心をなす例えから とったものだった。GNU のプログラムは「伽藍」で、感動的、集権的に計画化されたハッカー倫理のモニュメントであり、時の試練にも耐えられるように建てられたものだった。他方、 Linux は「沸騰する大バザール」にずっと似ていて、緩い分権的なインターネットの力学を通して開発されるソフトウェア・プログラムだった。レイモンドの論文は、 彼やストールマンや他の多くが使っていた伽藍方式を、特に GNU プロジェクトとストールマンに結びつけた。かくして、キャスティングは、ストールマンとトーバルズの比較としての開発方式の対照となった。ストールマン は、大聖堂建築家の古典的モデル -- つまり、18 ヶ月の間姿を消し、GNU C コンパイラみたいなものを携えて戻ってこれるプログラミングの「魔術師（wizard ; ウィザード)」-- で、トーバルズは、むしろディナーパーティの愛想のよいホストに近い。他の人達に Linux の設計に関する議論を先導させたり、テーブル全体でレフェリーが必要な場合だけ介入することで、トーバルズは彼自身のくつろいだパーソナリティをかなり反 映した開発モデルを作り上げた。トーバルズの視点から見れば、最も重要な運営上の役割は、制約を課すことではなく、アイデアの流れを維持することだった。

レイモンドは以下のようにまとめる。「ぼくはリーナスのいちばん賢く、影響力のあるハックは、Linux のカーネルそれ自体の構築ではなく、むしろ Linux 開発モデルの発明だったと思う。」4

これらの二つの開発方式の説明は鋭かったとしても、伽藍方式（それはレイモンド自身を含む、他の全ての人々が使っていたものだった）の連想をストー ルマンと特別に結びつけていたのは全くの中傷だった。事実、GNU Hurd を含む幾つかの GNU パッケージの開発者は、レイモンドの論文がしたようにさらに分析して広く代弁することはしなかったが、レイモンドが試みるより前にトーバルズのやり方につ いて読んでそれらを採用していた。レイモンドの論文を読んだ何千人という多くのハッカーが、この中傷で GNU に対する否定的態度に導かれたはずだ。

トーバルズの運営が成功した秘訣をまとめることで、レイモンド自身が、自由ソフトウェア・コミュニティ内の自由を優先事項にしない他のメンバーの関 心をひきよせることになった。彼らは企業の関心を自由ソフトウェアの使用と開発に向けさせるため、ビジネスにアピールする価値の観点から問題を取り上げる ことに決めた。強力で、信頼性が高く、安くて、先端的だということだ。レイモンドは、これらのアイデアの最も知られた提案者となり、彼らはネットスケープ （Netscape）の経営陣を獲得した。ネットスケープの占有的ブラウザは、やはり占有的ブラウザであるマイクロソフトのインターネット・エクスプロー ラに市場のシェアで負けつつあった。ネットスケープの役員は、レイモンドのスピーチに興味をそそられて、そのメッセージを本社に持ち帰った。数ヶ月後の 1998 年 1 月、ネットスケープは将来の開発にハッカーの支援を得られることを願い、主力製品のウェブブラウザ、ナビゲーター（Navigator）のソースコードを 公開する計画を発表した。

Netscape の CEO ジム・バークスデール（Jim Barksdale）が、会社の決定に大きな影響を与えたのはレイモンドのエッセイ、『伽藍とバザール』であると引用したとき、レイモンドは瞬く間にハッ カーのセレブの地位に昇格した。彼は何人かの人々を会合に招待した。それには、VA リサーチの創立者ラリー・オーガスチン（Larry Augustin）、出版社のオライリー・アンド・アソシエイツ（O'Reilly & Associates）の創立者、ティム・オライリー（Tim O'Reilly）、ナノテクノロジーを専門とするシリコンバレーのシンクタンクであるフォーサイト・インスティテュート（the Foresight Institute）の会長、クリスティーン・ピーターソン（Christine Peterson）が含まれていた。「会合の課題は一点に集約された。どうしたらネットスケープの決定を利用して、他の企業がこれに倣うようにさせられるだろうということだ。」

レイモンドは、そこで行われた会話は覚えていないが、最初に出てきた訴えを覚えている。フリーソフトウェアの「フリー」の言葉は自由を表していて価 格のことではない、と覚えてもらうためにストールマンや他のハッカーが最善の努力を尽くしても、そのメッセージはまだ浸透していなかった。たいていの企業 幹部は最初にフリーソフトウェアという用語を聞くと、フリーという言葉を「ゼロコスト」の同義語と解釈し、そこから先のメッセージにすぐ耳を貸さなくなっ てしまう。ハッカーがこの認識の不一致を切り抜ける方法を見つけるまで、ネットスケープによるソース公開が行われた後でさえも、自由ソフトウェア運動は高 い壁に直面していた。

自由ソフトウェアという大義の前進に積極的関心を持つ組織を率いるピーターソンは、別の用語を提案した。オープンソース（open source）である。当時を振り返り、宣伝業界で働く友達とネットスケープの決定について議論している間にオープンソースという用語を思いついたとピー ターソンは言う。その用語を思いついたのがどこからだったか別の分野から借りてきたのだったかは思い出せないが、その用語を友人が嫌ったことは覚えている。5

その会合では反応は劇的に違っていた、とピーターソンは言う。「私はそれを提案するのを躊躇していました、」とピーターソンは振り返る。「私はグ ループを代表する立場ではなかったし、それでオープンソースという言葉をさりげなく持ち出しました。新しい用語として強調するのではなくね。」ピーターソ ンには意外なことに、その用語は評判が良かった。会合が終わるまでに、レイモンドを含む参加者のほとんどが、その用語を喜んでいるように見えた。

オライリーが自由ソフトウェアについて語る会議の予定を立てたとき、モジラの始動パーティから一日二日経つまで、公開の場で、自由ソフトウェアに代 わる「オープンソース」という用語を使わなかったとレイモンドは言う。彼の会議を「フリーウェア・サミット」と呼んだオライリーは、メディアやコミュニ ティの注目を、ネットスケープにモジラを公開するよう促した他の価値あるプロジェクトにも向けさせたかったのだと言う。「こうした人達はみんな共通点が多 いのに、お互いに知らないので驚きました。」とオライリーは言う。

「すでに自由ソフトウェアの文化がいかに大きなインパクトを持っているのか世界に知らせ たいとも思いました。みんな自由ソフトウェアの伝統のかなりの部分を見逃していたんです。」

しかし、招待リストをまとめるにあたり、オライリーは長期的に政治的影響を及ぼすことになる決定をした。彼は招待リストを、Perl の作者のラリー・ウォール、sendmail の作者エリック・オールマン（Eric Allman）、BIND の作者ポール・ヴィクシー（Paul Vixie）ら、西海岸の開発者に限定することにした。もちろん、例外はあった。モジラの始動のため、すでに町にいたペンシルベニアの住人レイモンドはす ぐに招待された。ヴァージニア在住の Python の作者、グイド・ヴァン・ロッサム（Guido van Rossum）も同様だった。「うちの編集長で社内で一番 Python に詳しいフランク・ウィリソン（Frank Willison）が、まず僕に確認しないで彼を招待したんです」とオライリーは振りかえる。「彼が来てくれたのは嬉しかったけれど、始めは本当にローカ ルな集まりだったんですよ。」

推移を見守る人達には、招待リストにストールマンの名前を入れたがらないことで、彼を冷遇しているように見えた。「それでイベントに行かないことに した」とペレンズはサミットを振りかえって言う。一方、出席したレイモンドは、ストールマンを入れるのは無益だと主張したと言う。イベントのホストを務め るオライリーがソフトウェア・マニュアルの著作権の問題を巡ってストールマンと公然と争っているという事実のために、ストールマンを冷遇しているという噂 はさらに強まった。その会議よりも以前から、ストールマンは、自由ソフトウェアのマニュアルは自由ソフトウェアのプログラムと同様に、自由にコピーや修正 が可能であるべきだと主張していた。一方、オライリーは、自由でない本の付加価値市場は、より広範なコミュニティに自由ソフトウェアへのアクセスを可能に することでその有用性を高めていると主張した。二人はイベントのタイトルを巡っても対立していた。ストールマンは政治的な重みのない「フリーウェア」でな く、「フリーソフトウェア（自由ソフトウェア）」にこだわっていた。フリーウェアの用語は、たいてい、無料で使えるプログラムをさしていたが、それらは ソースコードが自由ではないために、自由ソフトウェアではなかった。

今振り返っても、オライリーはストールマンの名前を招待リストから外した決定を冷遇と見ない。「当時、僕はリチャードに直接会ったことは全然なかっ たんですが、電子メールのやり取りでは、彼は頑固で、対話に乗り気じゃありませんでした。僕はその会合で必ず GNU の流儀を代表してもらおうと思って、個人的に知っている、ジョン・ギルモアとマイケル・ティーマンを招待しました。GPL の価値に熱心でも、自由ソフトウェアのさまざまなプロジェクトやそれらの流儀の強みと弱みについて、快く率直な意見交換をやってくれるだろうと思ったから です。その後の大騒ぎを考えると、僕はリチャードも招待すべきだったのだと思いますが、僕がそうしなかったことを、GNU プロジェクトやリチャード個人に対する敬意が欠けていたからだと解釈すべきじゃないと思っています。」

冷遇があったにせよなかったにせよ、オライリーもレイモンドも、「オープンソース」と

いう用語は、サミット出席者から成功といえる十分な賛同を得た と言う。出席者はアイデアと経験を共有し、自由ソフトウェアのイメージを向上させる方法についてブレインストーミングを行った。主な関心は、マイクロソフ トの Windows に対する GNU/Linux の挑戦を強調することではなく、特にインターネットの基盤領域での自由ソフトウェアの成功を示す方法にあった。しかし、以前の VA での会合と同様、議論はすぐに「フリーソフトウェア」という用語が抱える問題になった。サミットのホストを務めたオライリーは、サミット参加者のトーバル ズのコメントを思い出す。

「リーナスはその当時シリコンバレーに引っ越してきたばかりで、『フリー（free)』という言葉が、英語では「自由（libre)」と「無料（gratis)」という二つの意味を持っていることをつい最近知ったばかりだと説明したんです。」

シグナスの創業者マイケル・ティーマンは、「フリーソフトウェア」という厄介な用語に替わる用語として、ソースウェア（sourceware）を提 案した。「誰も気乗りしなかった」とオライリーは振りかえる。「そのときにエリックが『オープンソース』という用語を口にしました。」

その用語は一部の気持をとらえたが、公式的な用語の変更を全員が支持するには遠かった。一日限りのカンファレンスの終わりに、出席者は三つの用語 -- フリーソフトウェア、オープンソース、そしてソースウェア -- を投票にかけた。オライリーによると、15 名の出席者のうち 9 名が「オープンソース」に投票した。まだこの用語にけちをつける人もいたが、参加者全員がこれから報道される議論ではそれを使用することで同意した。「連 帯のメッセージを携えて出て行きたかったんです」とオライリーは言う。

オープンソースという用語が語彙目録に入るまで時間はかからなかった。サミットから間もなく、オライリーは、ニューヨーク・タイムズ、ウォールスト リート・ジャーナル、その他有名どころの出版社の記者が出席する記者会見に、サミットの出席者を率いて行った。数ヶ月後に、トーバルズの顔がフォーブス誌 の表紙を飾り、ストールマン、Perl の作者のラリー・ウォール、アパッチ・チームのリーダーのブライアン・ベーレンドルフ（Brian Behlendorf）の顔が特集記事に現れた。オープンソースはビジネスにオープンだった。

ティーマンのようなサミット出席者にとって、連帯のメッセージこそ最も重要なものだった。彼の会社は自由ソフトウェアのツールやサービスを販売してかなりの成功を収めていたが、他のプログラマや経営者が直面する困難に気づいていた。

「フリー（自由/無料）という言葉の使用は疑いなく多くの状況で混乱をもたらしていました」とティーマンは言う。「オープンソースという言葉は、ビ ジネスフレンドリでビジネスに配慮する立場です。自由ソフトウェアは、道徳的に正しい立場。良かれ悪かれ、オープンソースの人達と足並みをそろえることは さらに利益になると計算しました。」

会議の後、新しい用語のオープンソースを知らせて、それを使うかどうかを聞くために、レイモンドはストールマンに電話をかけた。ストールマンはその 用語を採用するかどうかちょっと考えて結局それを捨てた。「僕はそれについて直接個人的に話したから知っている

んだ、」とレイモンドは言う。

ストールマンの直接の反応は、「ぼくはそれについて考えなきゃいけないね。」というものだった。その翌日、彼は、オープンソースという言葉による思 想の伝達をこれから確実にレイモンドとオライリーの価値観が支配するだろう、公衆の見解の中に自由なソフトウェアという理想を維持する最善の方法は伝統的 用語を捨てないことだ、と結論付けた。

1998 年の後半にストールマンは彼の立場を発表した。「オープンソース」は自由ソフトウェアの技術的利点を伝えるのには役立つが、話し手がソフトウェアの自由の 問題を消音することも助長した。それは意図しない「無料ソフトウェア」という意味と意図した「自由が尊重されるソフトウェア」という意味を等しく回避し た。後者の意味を運ぶ手段としては、従って、使いものにならなかった。実際には、レイモンドとオライリーは、コミュニティ内の非理想主義的政党に名前を与 えていたのだった。ストールマンはそれに同意しなかった。

加えて、ストールマンは「オープンソース」のアイデアは、ビジネスの支持の獲得の過大な強調に人々を導いていると考えていた。そのような支持はそれ 自身が必要悪ではないものの、有害な妥協に進んでしまうほどデスペレートにならないことを彼は期待していた。「Negotiation 101 は、誰かの同意を得るためにデスペレートになったときのことを教えてくれます、不利な取引を求めているのだ。」彼は言う。「ノーと言える用意があることが 必要です。」トーバルズ自らが「リナックス」コミュニティの「カミングアウト・パーティ」と宣伝するイベントだった 1999 年のリナックスワールド・コンヴェンション・アンド・エクスポで、ストールマンは自分の立場を要約して、仲間のハッカー達に、安易な妥協というルアーに負 けないように求めた。

「どれだけのことができるかを示してきたのだから、必死になって企業に協力したり、目標を妥協する必要はありません」と、パネルディスカッションで ストールマンは言った。「オファーしてくれれば受け入れるでしょうが、ぼくらのしていることを変えてまで、助けてもらおうとする必要はないのです。目標に向 かって一歩踏み出すことができるし、次の一歩またその次の一歩と歩んでいくことができます、そうすることで実際に目標に到達するでしょう。そうする代り に、中途半端な手段を講じて、次の一歩を踏み出さないこともできますが、それでは決して目標に到達しないでしょう。」

しかし、リナックスワールド・ショーの前でさえ、ストールマンはより融和的な仲間を遠ざけることにますます意欲を示していた。フリーウェア・サミッ トの数ヶ月後、オライリーは二年目の第二回 Perl カンファレンスを開催した。今回はストールマンも出席していた。自由ソフトウェアのアパッチ・ウェブサーバを商用製品に採用するという IBM の決定を称賛するパネルディスカッションで、ストールマンは聴衆に与えられるマイクを使い、Tcl スクリプト言語の作者であるパネリストのジョン・オウスターハウト（John Ousterhout）に鋭い非難を浴びせた。ストールマンは、オウスターハウトに自由ソフトウェア・コミュニティの「パラサイト」という烙印を押した が、それはオウスターハウトの創業した会社、

スクリプティックス社（Scriptics）を通じて Tcl の占有的ヴァージョンのマーケティングをしていたからだ。オウスターハウトは、スクリプティックス社は、Tcl の自由ヴァージョンには、ごく最小限の改良にのみ貢献していくことになると述べていた。それは実際には、小さな貢献を使って大規模な自由ではないソフト ウェア開発へのコミュニティの承認を勝ち取ろうということだった。ストールマンは、この立場を退けてスクリプティックス社のプランを非難した。「Tcl が引き続き存在していくのにスクリプティックス社が必要だとは思わない」とストールマンが発言すると、聴衆の仲間達から制止された。6

「あれはかなり見苦しい光景でした、」とプライム・タイム・フリーウェアのリッチ・モーリンは振りかえる。「ジョンは、Tcl や Tk や Sprite でとても尊敬すべきことを成し遂げています。彼は本物の貢献者です。」ストールマンとストールマンの立場には共感するが、モーリンはストールマンの売り言 葉に悩まされた人達のほうに同情した。

ストールマンは謝ったりしない。「占有的ソフトウェアを批判することは見苦しいことではありません。占有的ソフトウェアこそ、見苦しいのです。オウ スターハウトが過去に本物の貢献をしたのはその通りですが、ポイントはスクリプティックス社がほとんど百パーセントの占有的ソフトウェア会社になろうとし ていたことでした。あのカンファレンスでは、自由を支持することは、ほぼ全員に異を唱えることを意味していました。聴衆としての発言なので、ぼくは少し話 すことができませんでした。すぐ忘れられないように問題をとりあげる唯一の方法は、それを強い言葉で言うことでした。」

「トーバルズが瑣末な事柄について敵意のある事を言うのを「感心できないが面白い」と言いながら、ぼくが誰かの行動について真剣な批判をするときに『もめ事を起こす』と言って非難する人達は、ダブルスタンダードを適用しているように思います。」

ストールマンの物議を醸した批判は、一時的に、潜在的支持者のブルース・ペレンズを遠ざけることになった。1998 年にエリック・レイモンドは、「オープンソース」という用語の使用を監視し、自社製プログラムをオープンソースにすることに関心を持つ企業にオープンソー スの定義を提供する組織、オープンソース・イニシアティブ（OSI）を立ち上げることを提案した。レイモンドは、その定義の起草者にペレンズを起用した。 7

ペレンズはその後、OSI がストールマンと FSF に対立して設立されたことに遺憾の意を表明して OSI を辞任することになろう。それでも、自由ソフトウェア財団の後援を受けない自由ソフトウェアの定義の必要性を振り返り、ペレンズは他のハッカーがいまだに 距離を置く必要を感じるかもしれない理由が分かる。「リチャードは本当に好きだし素晴らしいんだ。」とペレンズは言う。「リチャードにもっとバランスが あったら、彼の仕事はもっとうまくいくのに。二ヶ月ぐらい自由ソフトウェアから離れたっていいんだよ。」

ストールマンの力では、オープンソース支持者の宣伝の勢いにほとんど対抗できなかった。1998 年 8 月、半導体メーカーのインテルが GNU/Linux ベンダーのレッドハットの株を購入したときに、それを報じたニューヨーク・タイムズの記事は、レッドハットを「自由ソフトウェアあるいはまたオープンソー スとして知られる運動」の成果と書いた。8 半

年後、ジョン・マーコフ（John Markoff）のアップル・コンピュータの記事は、見出しで、アップルは「オープンソースの」アパッチ・サーバを採用と発表した。9

こうした勢いは、「オープンソース」という用語を積極的に利用した会社の増大する勢いと同時にすすんだ。1999 年 8 月に、自らを「オープンソース」として熱心に売り込んでいた会社のレッドハットが、ナスダック（Nasdaq）に上場した。その年の 12 月、VA リナックス -- かつての VA リサーチ -- の新規株式公開が歴史的な効果をもたらした。1 株 30 ドルで始まった VA リナックスの株価は、最初の取引で爆発的に値上がりして 300 ドルを突破し、終値でも 239 ドルの水準に戻しただけだった。好運にも VA 株を最も安く手に入れ、取引終了まで持ち続けた株主は、紙の上の資産としては 698％ の増加を体験することになった。ナスダックの新記録だった。エリック・レイモンドは、取締役会のメンバーとして、三千六百万ドル相当の株を所有していた。 しかし、これらの高値は束の間のものだった。それはドットコム・バブルの終わりとともに崩壊した。

オープンソース支持者のメッセージは簡単だった。自由ソフトウェアのコンセプトを売りこむのに必要なのは、それを事業主に優しいものにすることだ。 彼らにはストールマンと自由ソフトウェア運動は市場と闘っているように見えた。彼らはその代わりに、それに投機させることを追求した。高校の仲間外れの役 割を演じるのではなく、彼らはセレブのゲームを演じ、その過程で力を強めていった。

これらの方法はオープンソースに大成功をもたらしたが、自由ソフトウェアの理想にではなかった。「メッセージを広める」ためにしたことは、その最も 重要な部分、倫理的問題としての自由の理念を省略することだった。この省略の結果は、今日、目に見えるものになっている。2009 年現在、ほとんどすべての GNU/Linux ディストリビューションは占有的プログラムを含んでいて、Linux のトーバルズのヴァージョンは占有的ファームウェアを含んでいる。そして、以前、VA リナックスと呼ばれた会社は占有的ソフトウェアをベースにビジネスをしている。世界のウェブ・サーバーの過半はアパッチのいずれかのヴァージョンを走らせ ていて、アパッチの普通のヴァージョンは自由ソフトウェアだが、それらの多くのサイトで IBM が配布している占有的に修正されたヴァージョンが走っている。

「彼の最悪の時には、リチャードはリーナス・トーバルズと僕が共謀して、彼の革命を乗っ取っとろうとしてると思い込んでいたんだ」とレイモンドは言 う。「オープンソースという言葉に対するリチャードの拒絶と、僕の見解の中に彼が入念に持ち込んだイデオロギー的な溝は、理想主義と縄張り意識の奇妙な混 合から来ている。世の中には、それはみんなリチャードの個人的なエゴだと信じる人もいる。僕はそう思わない。それよりも、自由なソフトウェアという理念と 自分自身を結びつけすぎて、その理念を脅かすものは何でも彼自身を脅かすものに見えるんだ。」

ストールマンは次のように答えている。「レイモンドは、ぼくの見解を正確に言わない。ぼくはトーバルズが誰かと「共謀」すると思いません。こそこそ したやり方は彼のやり方じゃないからね。それでも、レイモンドの汚らわしい振る舞いは、彼の言動そもものから明

らかです。功罪についての僕の意見に答える よりも（彼がそう主張するときでさえ)、彼はそれらの心理学的解釈を提案します。彼は匿名の誰かのものだというどぎつい解釈を考え、次に、それより少し軽 蔑的なものを提案してぼくを「弁護」します。彼はこのやり方でよくぼくを「弁護」してきました。」

皮肉にも、オープンソースとレイモンドのようなオープンソース主唱者の成功は、ストールマンのリーダーとしての役割を衰退させなかった。しかし、そ れは彼が何のリーダーなのかについて、多くの人々を誤解させることになった。自由ソフトウェア運動は、オープンソースのように企業やメディアから認知され ていなかったので、GNU/Linux の大半のユーザーはその見解はおろか、その存在さえ聞いたことがなかった。彼らはオープンソースのアイディアや価値について聞いていたが、ストールマンに 別の見解があるとは思ってもみなかった。かくして、彼は「オープンソース」を擁護してくれたことに感謝するメールを受け取ることになるが、返信で彼はそれ の支持者になったことはないことを説明し、自由ソフトウェアについて送り主を啓蒙する機会にした。

ライターによっては「FLOSS」という用語を使うことで、「自由ソフトウェア」という用語を受け入れている。FLOSS は、「自由（Free/Libre）とオープンソース（Open Source）のソフトウェア（Software)」を表す。しかし、彼らは、しばしば一つの FLOSS 運動があると言う。それはアメリカには一つの「リベラル/保守」運動があると言うのに似ている。そして、この想像上の一つの運動に関連付けられている見解 は、たいてい、彼らの聞かされてきたオープンソースの見解である。こうしたいろいろな障害はあるが、自由ソフトウェア運動もたまにはその思想が聞き届けら れ、絶対的な意味では成長を続けている。信念を守り、オープンソースと対比してその思想を提示して確実な地歩を得ている。「ストールマンの性格の基本的な 特徴の一つは意見を変えないことです」とイアン・マードックは言う。「もしそれだけの時間が必要なら、人々が彼の考え方に同調するまで十年でも待つのです。」

マードック自身は、その譲らぬ気質に励まされるし貴重なものを見いだす。ストールマンはもう自由ソフトウェア運動の孤独な指導者ではないかもしれな い。だが、今でも自由ソフトウェア・コミュニティの北極星だ。「彼が自分の考え方をこの先も一貫していくことは、あなたもおわかりでしょう」とマードック は言う。「たいていの人はそうはいかない。賛否はともかく、あれは本当にリスペクトしないといけません。」

後注

1 ピーター・サルス『FYI‐ 自由に再配布可能なソフトウエア、２月２日、ケンブリッジ』（1995）を参照。（テリー・ウィノグラッドによりアーカイブ）

http://bat8.inria.fr/~lang/ hotlist/free/licence/fsf96/call-for-papers.html

2 リーナス・トーバルズはフィンランド人だが、母語はスウェーデン語である。『The Rampantly Unofficial Linus FAQ』に簡潔な説明がある。

フィンランドには、無視できない数(約 6%)のスウェーデン語を話すマイノリティ人 口が

ある。彼らは自らを「finlandssvensk」あるいは「finlandssvenskar」と呼び、自分たちをフィンランド人だと考えてい る。彼らの家族の多くは、数世紀に渡ってフィンランドで暮らしてきた。スウェーデン語は、フィンランドの二つの公用語の一つである。

3 ブルックスの法則とは、ブルックスの著作の次の引用を手短に要約したものだ。

ソフトウェアの構築は、本来体系的な活動 -- 複雑な相互関係を伴う実践 -- であり、物事を伝えるための労力も並大抵ではないので、分業がもたらす個々人の作業時間をすぐに食い潰すようになる。だから、増員はスケジュールの短縮ではなく遅延をもたらす。

フレッド・P・ブルックス『人月の神話』(Addison Wesley Publishing, 1995)を参照。

4 Eric Raymond, "The Cathedral and the Bazaar" (1997) を参照。日本語訳、エリック・レイモンド『伽藍とバザール』http://cruel.org/freeware/cathedral.html

5 Malcolm Maclachlan『儲け主義がオープンソース運動を分裂させる』TechWeb? News(1998 年 8 月 26 日)を参照。

http://www.techweb.com/article/showArticle? articleID=29102344

6 前掲書

7 ブルース・ペレンズ他『オープンソースの定義』オープンソース・イニシアティブ(1998年)を参照。 日本語訳は、http://www.opensource.jp/osd/osd-japanese.html

http://www.opensource.org/docs/definition.html

8 アミー・ハーモン『売ります:自由なオペレーティング・システム』ニューヨーク・タイムズ(1998 年 9 月 28 日)を参照。

http://www.nytimes.com/library/tech/98/09/biztech/articles/ 28linux.html

9 ジョン・マーコフ『アップル、サーバー・コンピュータに「オープンソース」を採用』ニューヨーク・タイムズ(1999 年 3 月 17 日)を参照。

http://www.nytimes.com/library/tech/99/ 03/biztech/articles/17apple.html

第十二章　ハッカー地獄巡りの小旅行

［RMS: この章の私の変更は、このラベルをつけた幾つかの注釈だけである。］

リチャード・ストールマンは、キヘイのダウンタウンを通り抜ける間ずっと、信号が変わるのを待ちながら、瞬きもせずレンタカーのフロントガラス越しに凝視している。

私たち二人は、近くにあるパイアの町に向かっている。そこで、何人かのプログラマとその妻たちと一緒にディナーを食べるため、約 1 時間後に落ち合う予定だ。

マウイ高性能計算センターのストールマンの講演から約 2 時間、講演前はとても感じの良かったキヘイの町が、今やずいぶん非協力的に感じられた。ビー チの都市はたいていそうだが、キヘイは、郊外に細長く拡がっている。ハンバーガー屋、不動産屋、ビキニ・ショップが際限なく続くメイン・ストリートを車で 走っていくと、商業主義の巨大サナダムシの消化管を通過して行く、金属コートされた一片の食物のように感じずにはいられない。側道が無いことが、さらに気 分を悪くする。前以外には進みようがなく、車はバネのように揺すられながら進んで行く。200 ヤード先で信号が青に変わる。私たちがそこにたどりつく頃に 信号はまた黄色へと変わる。

ずっと東海岸に住んできたストールマンにとって、晴れたハワイの素敵な午後の時間を交通渋滞にはまって過ごすという見通しは、塞栓症を引き起こすの に十分だ。［RMS: 私は運転をしていたし、メールの返事をする時間もなくなり、やっと我慢していたが本当に苦痛だった。］素早く右折を繰り返して 4 分の 1 マイル戻れば、この状況全てを簡単に避けられることを知っているから、なお悪い。不運なことに、私たちは前を走る車の運転 手の意のままだ。この運転手は道を知っている研究所のプログラマで、隣接するピラニ・ハイウェイではなく、景色のよい道を通って私たちをパイアに連れて行 こうと決めていた。

「こりゃ、ひどい。」いらだちのため息の間に、ストールマンは言う。「どうしてもう一方のルートにしなかったんだ？」

再び、4 分の 1 マイル先の信号が青に変わる。再び、車数台分の距離をのろのろと進む。こんなプロセスがさらに 10 分ほど続いて、やっと隣のハイウェイにアクセスできるはずの大きな交差点にたどり着く。

前の車のドライバーはそれを無視して、交差点を通過する。

「どうして曲がらないんだ？」ストールマンは呻き、いらいらして腕を振り上げる。「信じられるかい？」

私はどちらにも答えないことにする。ここマウイで、ストールマンが運転する車に同乗していること自体が十分信じられない話だ。2 時間前には、ストールマンが運転の仕方を知っていることさえ知らなかった。今は、カーステレオから流れてくるヨーヨー・マのチェロ演奏「アパラチアン・ ジャーニー（Appalachian Journey）」の悲しげな低音の響きに耳を傾け、左手の日没を眺めながら、私は車の内装に溶け込んでしまおうと懸命だ。

ついに道を曲がる次のチャンスがやってくると、ストールマンは、前の運転手に合図しようとして右折の指示を出す。うまく行かない。またもや、交差点をのろのろと通過し、次の信号の 200 ヤード手前で止まる。今や、ストールマンは激怒している。

「ぼくたちをわざと無視しているみたいだ。」と彼は言い、飛行機の地上誘導員みたいなジェスチャーやパントマイムをして、ガイドの目をひこうとするが無駄だ。ガイドは動じなかったようで、次の 5 分間、私たちには、バックミラーに写る彼の頭がちょっと見えるだけだ。

ストールマン側の窓を見る。カホオラウェ島とラナイ島付近は、日没を理想的な額縁に切り取ってくれる。息を呑むような光景で、もしハワイ人なら、こ んな時間をもうちょっと辛抱させてくれる光景だと思う。ストールマンにそちらに注意を向かせようとするが、今、ストールマンは前の運転手の注意を払わない 態度に頭がいっぱいで知らんぷりだ。「ピラニ・ハイウェイ　次を右折」の表示を完全に無視して、運転手が青信号をもう一つ通過するとき、私は歯軋りする。 BSD プログラマのキース・ボスティックから早くから伝え聞いていた警告を思い出す。「ストールマンは、愚かさによろこんで耐えたりしない、」とボスティックは 私に警告した。「誰かが何かまぬけなことを言ったりしたりすると、彼は目を見て言うんだ、『それは愚かだ』。」

前方の鈍感な運転手を見て、今ストールマンに苦痛を与えているのは、融通がきかないことではなく、愚かさだということに気づく。

「彼はまるで、そこに行くのにどうすれば効率的かをまったく考えずに、経路を選んだみたいじゃないか」と、ストールマンは言う。

「効率的」ということばが、悪臭のように車中を漂う。非効率性ほど、ハッカーの心を苛立せるものはない。ストールマンに、最初にプリンタのソース・ コードを調べようと思わせたものは、ゼロックスのレーザープリンタを、日に 2 度も 3 度もチェックすることの非効率性だった。ストールマンに、シンボリクス社と戦わせ、GNU プロジェクトを開始させたのは、商用ソフトウェア・ベンダに乗っ取られたソフトウェア・ツールを書き換えることの非効率性だった。もし、ジャン・ポール・ サルトルがかつて言ったように、他人が地獄なのであれば、ハッカーの地獄とは他人の愚かな誤りを繰り返すことであり、ストールマンの全人生は、こうした地 獄の業火から、人類を救済する試みだ、と言っても過言ではない。

ゆっくりと通りすぎていく風景を見ていると、この地獄の喩えは、いよいよ明白だ。多くの店や駐車場や上手く時間調整されてない街路灯によって、キヘ イの街は、都市というより、下手に設計された大規模ソフトウェア・プログラムのようだ。交通の経路を変えて、車の流れを脇道や高速道路に分散させるのでは なく、都市計画担当者は、一本のメイン・ストリートにすべてを通させることを選んだのだ。ハッカーの観点からすれば、この大混乱のただ中で車に座っている のは、爪で黒板をひっかく音の CD を大音量で聴いているようなものだ。

「不完全なシステムは、ハッカーを激怒させる、」とはスティーブン・レヴィの観察だが、

これは、ストールマンとともに車に乗り込む前に聞いておくべ きもうひとつの警告だった。「ここに、ハッカーが一般にクルマの運転を嫌うもうひとつの理由がある -- ランダムにプログラムされた赤信号やへんてこに配置された一方通行からなる交通体系は、忌々しくも不必要な［スティーブン・レヴィによる強調］遅延を引き 起こすから、標識を再配置し、交通信号の制御ボックスを開き ... 全体系を再設計したい衝動に駆られてしまう。」1

とはいえ、もっといらいらさせるのは、信頼をおいているガイドの不誠実さだ。賢い近道を探し出す -- 本物のハッカーなら誰でも本能的にやることだ -- のではなく、私たちの前を行くドライバーは、都市計画者のゲームにずっとつき合うことを選んでいる。ダンテの『地獄』のヴェルギリウスのように、ガイド は、私たちが望もうが望むまいが、ハッカー地獄の完全ガイドつきツアーをやろうと決意しているのだ。

この所見をストールマンに話すより前に、くだんの運転手はついに右折の合図を出す。ストールマンの力んだ肩が少しほぐれ、一瞬、車内の空気から緊張 が抜ける。だが、前の運転手が速度を落とすと、緊張が戻ってくる。「この先工事中」の標識が道の両側にあり、ピラニ・ハイウェイまであと 4 分の 1 マイルもないのに、僕らとハイウェイを結ぶ二車線の道は、休眠中のブルドーザーと二つの巨大な土砂の山で塞がれている。

我らがガイドが目の前で不器用な五角 U ターンを始めるとき、ストールマンは、何が起きているのか理解するのに数秒かかる。ブルドーザーとちょうどその向こうの「通り抜けできません」の標識をちらっと見るに及んで、ついにストールマンの堪忍袋の緒が切れる。

「なんで ? なんで ? なんでなんだよ ?」彼は泣きそうな声を出し、のけぞる。「道が閉鎖されているなんて分かったはずだろう。この道が使えないことは分かったはずだよ。わざとやったんだ。」［RMS: 私は、彼がわざと遅い道を選んでいることを言っていた。次に説明するように、これらの引用は正確ではないと思う。］

運転手は、方向転換を終え、僕らの脇を通りすぎて、メイン・ストリートに戻る。彼はそうしながら、頭を振って我々にすまなそうに肩をすくめて見せ る。歯の見えるにこやかな笑顔と合わさって、運転手のジェスチャーは本土人的いらだちを表してはいるが、それは島の人間の運命論の保護で調節されている。 レンタカーの閉じた窓ごしに、簡潔なメッセージが読み取れる。「ヘイ、ここはマウイさ。しかたがない。」

ストールマンはもう我慢できない。

「笑うな！（"Don't you fucking smile!"）」彼は怒鳴り、窓ガラスが曇る。「きみのせいだろ（It's your fucking fault.）。ぼくのやり方でやっていれば全部もっと簡単だったんだ。」［RMS: 私は副詞に "fucking" を使わないから、これらの引用は不確かなようだ。これはインタヴューではないから、ウィリアムズはテープレコーダーを使っていなかっただろう。起きたこと は大体彼が書いたようなことに違いないが、これらの引用は、たぶん、私の言葉より彼の理解を反映している。］

ストールマンは、ハンドルを握り、それに二度身を引き寄せ、「ぼくのやり方」という言葉にアクセントをつける。ストールマンが身を揺さぶる様子は、 クルマの座席でかんしゃ

くを起こした子どものそれに似ている。このイメージはストールマンの声の調子で強調される。怒りと苦痛の間で、ストールマンは泣き 出しそうだ。

運良く涙は出ずにすむ。夏の夕立のように、かんしゃくは始まると同時に収まる。愚痴っぽく二、三度あえぎ声を出した後で、ストールマンはギアをバッ クに入れ、自分のUターンを始める。メイン・ストリートへ戻るまでに、彼の表情は 30 分前にホテルを出たときのように冷静になっている。

次の交差点までは、五分もかからない。そこからは簡単にハイウェイにアクセスでき、たちまち、パイアに向ってリラックスできるスピードで疾走し始め る。ストールマンの左肩に明るい黄色で迫っていた夕陽は、今はバックミラーのなかでクールなオレンジ・レッドに燃えている。太陽は、ハイウェイ両側で僕ら から飛び去って行く二列のウィリウィリの木をその色に染めている。

次の 20 分間、車のエンジンやタイヤまわりのうなり音を除けば、ぼくらのクルマで響いている音は、アパラチアン・フォークの調べのもの悲しげな緊張感を奏でるチェロとバイオリンの三重奏の音色だけだ。

後注

1. スティーブン・レビー著『ハッカーズ』(工学社, 1987/02)参照。

第十三章　戦い続ける

リチャード・ストールマンにとって、時はすべての傷を癒してくれないかもしれないが、良い味方を与えてくれる。

『伽藍とバザール』の 4 年後、ストールマンは、レイモンドの批判にまだ苛立っていた。リーナス・トーバルズが世界で最も有名なハッカーの役に昇格したことも不満だった。彼は、1999 年頃の Linux のトレードショーに登場して人気を博した T シャツを思い出す。スター・ウォーズの宣伝ポスターを真似たデザインの T シャツは、トーバルズがルーク・スカイウォーカーのようにライトセーバーを振りかざし、ストールマンの顔が R2D2 に乗っていた。それがストールマンの気に障るのは、彼をトーバルズの助っ人として描いているだけでなく、トーバルズを自由ソフトウェア・コミュニティと オープンソース・コミュニティのリーダー役に持ち上げていたからだ。その役目はトーバルズ自身が受け入れたがらないものだ。「皮肉なことに、」とストール マンは嘆く。「その剣をとることこそ、リーナスの拒んでいることですよ。みんなが彼を運動のシンボルと見るようになった。そして、彼は戦わない。どこにい いことがあるんでしょうか。」

そう思っていると、ストールマンの、ハッカー・コミュニティにおける倫理的裁定者としての評判を高める道を開いたのは、トーバルズの「剣をとるこ と」への消極性だった。不満があったとしても、この数年間は、彼にも彼の組織にも大変よかったことをストールマンは認めるべきだ。ユーザーがそれを "Linux" として考える GNU/Linux システムの皮肉な成功により脇に追いやられたが、それでも、ストールマンは再びイニシアチブをとることに成功した。2000 年 1 月から 2001 年 12 月までのストールマンの講演スケジュールには、六大陸に立ち寄り、ソフトウェアの自由の概念が重く響く国々 -- 例えば、中国やインド -- を訪れることが含まれていた。

公職の権威である外にも、ストールマンは、彼が管理人を勤める GNU 一般公衆ライセンス（GPL）の梃子の力を巧みに活用してきた。1999 年の Linux 関連会社の新規株式公開バブルが急速にしぼんでいった 2000 年の夏、ストールマンと自由ソフトウェア財団は、二つの大きな勝利を収めた。2000 年 7 月に、ノルウェーのソフトウェア企業で、Qt という、GNU/Linux オペレーティング・システムにとって有用な統合グラフィックツールの開発を行っていたトロールテック社 (Trolltech) が、同社のソフトウェアを GPL でライセンスすると発表した。数週間後、そのときまで、同社のソフトウェア所有権の完全な支配を放棄しないまま、オープン・ソースという時流に抜目なく乗 ろうとしていたサン・マイクロシステムズは、ついに折れて、新しいオープンオフィス（OpenOffice?） 統合ソフトウェアを、LGPL（劣後 GPL;the Lesser GNU Public License）と、サン産業標準ソースライセンス(SISSL;the Sun Industry Standards Source License)の、二本だてのライセンスにすることを発表した。

トロールテック社の場合、勝利は GNU プロジェクトの長期にわたる努力の結果だった。

Qt が自由ソフトウェアでないことは、自由ソフトウェア・コミュニティにとって深刻な問題だった。人気を得つつあった自由なデスクトップ環境の KDE が Qt に依存していたからだ。Qt は自由ではないソフトウェアだったが、トロールテック社は、自由ソフトウェアのプロジェクト（KDE のような）をそれを無料で使うために招いていた。KDE 自身は自由ソフトウェアだったが、自由を強く主張しているユーザーは、Qt をはねつける必要があったので、それを使うわけにはいかなかった。ストールマンは、多くのユーザーが GNU/Linux でグラフィカルなデスクトップを使いたいであろうこと、そして、中に Qt が隠れていても、大半は KDE の誘惑をはねつけるほど自由に重きを置かないだろうことを認識していた。

この危険に取り組むために、ストールマンは、並行的な二つの対抗プロジェクトを開始するために人々を募った。一つは GNOME、GNU 自由グラフィカル・デスクトップ環境だった。もう一つは Harmony、互換性のある Qt の自由な取替プログラムだ。もし GNOME が成功すれば KDE は不要になる。もし Harmony が成功すれば Qt は不要になる。どちらにせよ、自由ではない Qt がなくてもユーザーは GNU/Linux のグラフィカル・デスクトップ環境を持てることになる。

1999 年に、これら二つの努力は順調に進み、トロールテック社の経営陣はプレッシャーを感じ始めていた。そのため、トロールテック社は自社の自由ソフトウェアラ イセンス、QPL で Qt をリリースした。QPL は自由なライセンスとして認められたが、ストールマンは GPL との非互換性という欠点を指摘した。一般に、GPL でカバーされたコードを一つのプログラムの中で Qt と結びつけようとするのは、どちらかのライセンスを破らないと不可能だった。結局、トロールテック社の経営陣は彼らの目的に GPL は等しく役に立つことを認めて、Qt を二重ライセンスの下でリリースした。同じ Qt のコードが、同時に、GNU GPL でも QPL でも使えるようになった。三年後の、これは勝利だった。

Qt が自由になると、Harmony（それはまだ実際の使用には不十分だった）を開発する動機は消滅して、開発者はそれを放棄した。GNOME は十分な勢いを獲得していたため、開発が続けられ、主要な GNU グラフィカル・デスクトップであり続けている。

サンの場合は、自由ソフトウェア財団の条件に従ってプレーすることを望んだ。1999 年、オライリーのオープンソース・カンファレンスで、サン・マイクロシステムズの共同設立者でチーフ・サイエンティストのビル・ジョイは、自社の「コミュ ニティ・ソース」ライセンスを擁護していた。本質的に水で薄めた譲歩だったこのライセンスは、ユーザーに対してサン所有のソフトウェアをコピーし改変する ことを許すが、サンと交渉して使用料について合意しない限り、そのソフトウェアで料金請求することを認めなかった（この制限のため、ライセンスは、自由な ライセンスとしても、オープンソースのライセンスとしても、認められなかった）。ジョイのスピーチから 1 年後、同じ舞台に現れたサン・マイクロシステムズの副社長マルコ・ボエリーズ（Marco Boerries）は、ライセンス上の新たな譲歩を説明した。それは特に GNU/Linux オペレーティング・システムのために設計した統合オフィス・

アプリケーションのオープンオフィスに適用されるものだ。
「それは、3 文字で綴れます。」とボエリーズは言った。「GPL です。」
この時ボエリーズは、同社の決定はストールマンとほとんど関係がなく、GPL で保護されたプログラムの勢いによるところが大きいと言った。「それぞれ異なる製品は、それぞれ異なるコミュニティをひきつけるという基本認識が生じたの です。だから、どんなライセンスを使うかは、どんなコミュニティをひきつけたいかによります。」と、ボエリーズは言った。「[オープンオフィス] に関しては、GPL コミュニティと一番関連があることは明らかでした。」2 惜しいかな、この勝利は、不完全だった。オープンオフィスは自由ではないプラグインの使用を推奨していたからだ。
このようなコメントは、GPL の隠れた力を示すもので、間接的に、その創造に最大の役割を果たした男の政治的天才を示している。「GPL を今あるような姿で起草する弁護士なんて、まずいません、」とコロンビア大学の法学教授で自由ソフトウェア財団の法律顧問のエベン・モグレンは言う。「で も、うまく機能してるんです。それがうまくいくのは、リチャードの設計哲学のおかげです。」
かつて職業プログラマだったモグレンがストールマンとしている無料奉仕活動 (pro bono) は、ストールマンがある個人的事件でモグレンに法的援助を求めた 1990 年までさかのぼる。当時モグレンは、連邦政府と法的紛争の真最中だったジマーマンと仕事をしていたが、ストールマンの依頼を光栄に思ったと言う。3
「彼に言ったんですよ。毎日 Emacs を使って暮らしているから、僕の方も、とんでもない量の弁護士業務で借りを返さないとね、って。」
それ以来、モグレンは、おそらく他の誰にも増して、ストールマンのハッカー哲学が法律の領域と交差するところを観察する絶好のチャンスに恵まれた。 モグレンによれば、ストールマンの法律に対するアプローチの仕方とソフトウェアのコードに対するアプローチの仕方は大体同じだ。「弁護士の立場からする と、法的文書でするべきことが、すべてのバグを取り除くことだという発想はあまり意味がないと言うべきです。」とモグレンは言う。「どんな法的プロセスに も不確実性があります。そこで、たいていの弁護士がしたがるのは、自分の依頼者に有利な形で不確実さを記録しておくことです。リチャードが目指すのは、まさにその正反対ですね。彼の目標は不確実性を取り除いてしまうことですが、それは本来、不可能なことです。世界中の全ての法体系で、全ての環境をコント ロールできるような一つのライセンスを起草するなんてことは本来不可能です。でも、そうするとしたら、彼のようにするしかありません。その結果として生ま れる設計のエレガントさ、設計の単純さが、達成すべきことをほとんど達成してしまうんです。弁護士は、あとちょっと手を加えるだけで、仕事ははるか先まで 進むんです。」
ストールマンの課題の推進役として、モグレンは、味方になるつもりの人間の欲求不満を理解する。「リチャードは、自分が根本的問題だと考える事柄を 妥協したがらない男です」とモグレンは言う、「それに、言葉を曲げることはおろか、巧くぼかすことさえ、簡単に受

け入れない。言葉をぼかすぐらい、人間社 会がしばしば多くの人に求めるものなんだけれどね。」

自由ソフトウェア財団の支援に加えて、モグレンは、たとえばドミトリ・スクリャロフ（Dmitri Skylarov）や DVD 暗号解読プログラムの deCSS の配布者のような、別の著作権の被告たちに法的支援を提供している。

スクリャロフは、ロシアの会社の従業員として、アドビ社（Adobe）の電子書籍のデジタル・コピー防止を破るプログラムを書いてリリースしたが、 ロシアでそれに反対する法律はなかった。そして、彼は仕事の科学論文について話すためにアメリカを訪れたときに逮捕された。ストールマンはアドビがスク リャロフを逮捕させたことを非難する抗議に熱心に加わり、自由ソフトウェア財団はデジタル・ミレニアム著作権法を「ソフトウェアの検閲」だとして公然と非 難したが、スクリャロフのプログラムに賛成して介入するわけにはいかなかった。それは自由なものではなかったからだ。かくして、モグレンは電子フロンティ ア財団を通してスクリャロフの弁護のために働いた。自由ソフトウェア財団は、deCSS の配布に関わるのを避けた。それは非合法だったからだが、ストールマンは deCSS を禁じているアメリカ政府を非難し、そしてモグレンは被告の法廷弁護人として働いた。

それらの事件では、自由ソフトウェア財団自身は関わらないという自由ソフトウェア財団の決定の後で、モグレンは、ストールマンの頑固さの価値を正当 に評価することを学んだ。「この数年間に何度も、ぼくはリチャードに言いに行ったんです、『ぼくらはこれをするべきだ。あれをすべきだ。これは戦略的に重 要な状況だ。次の手はこれだ。きみがしなくちゃいけないことがあるんだ。』すると、リチャードの返事はいつもこうでした、『ぼくらは何もする必要がない よ。』ただ、待ってればいい。する必要があることは、できている。」

「で、どうなったと思いますか？」とモグレンは付け加えた。「たいてい、彼は正しかった。」こういうコメントは、ストールマン自身の自己申告を否定 している。「ぼくは、ゲームは得意じゃありません。」彼を抜け目のない戦略家と見る多くの見えざる批評家に、ストールマンは言う。「ぼくは先を見越した り、人が将来するかもしれないことを予測するのは得意じゃないんです。ぼくのアプローチはいつも、財団[の理想]に焦点を合わせることでした。『ぼくらに できる限り、財団を強化していこう。』と言っていました。」

GPL の人気の拡大と GPL の引力が強まり続けていることは、ストールマンと GNU の仲間たちが築いた財団への最大の贈物だ。ストールマンは、決して自由ソフトウェアをリリースする世界唯一の人ではなかったが、自由ソフトウェア運動の倫 理的枠組みを作った唯一の功労者だった。現代のプログラマがその枠組みの中で働いて快適に感じているかどうかではない。彼らが選択肢をもったこと自体が、 ストールマンの最大の遺産なのだ。

この時点で、ストールマンの遺産を議論するのはやや時期尚早だ。この執筆時点で 48 歳のストールマンには、まだ、この遺産に追加しまたは差引きするだけの年月の猶予がある。それでも、自由ソフトウェア運動の勢いは、ソフトウェア産業との 日々の闘いを離れ、そして、より威厳に満ちた歴史的背景の中で、ストールマンの人生を検証する誘惑に誘う。

彼の名誉のために言えば、ストールマンは、この話題でどんな当てずっぽうを語ることも拒否する。「未来がそうなって行くような詳細な計画を練るなん てことは決してできないんです。」と、ストールマンは言い、まだ尚早な墓碑銘を提案する。「『ぼくは戦い続ける。どこまで行けるか誰が知ろう。』としか言 えなかった。」

戦うときに、ストールマンが自分の最大の味方になったかもしれない人々、彼自身の見解ではなく彼らの見解のために闘いたい人々を、離反させてきたこ とに疑問の余地はない。だが、ストールマンのかつての政敵の多くが、話を向けられると、彼に対して褒め言葉を発しようとすることは、彼のまっすぐで倫理的 な性格を証明している。しかし、イデオローグのストールマンと天才ハッカーのストールマン。この間にある緊張は伝記作者にこんなことを考えさせる。ストー ルマンの人柄がもう関係なくなったら、人々のストールマン観はどんなものになるのだろうか？

本書の初期の草稿で、私はこの問題を「100 年」問題と呼んでいた。ストールマンと彼の仕事を客観的に見るように望んで、ソフトウェア産業の多様な著名人に、現在の時間枠から抜け出だして、100 年後の未来から自由ソフトウェア運動を振り返る、歴史家の立場に立ってくれるように頼んだ。現在の見通しの良い地点から、ストールマンと過去のアメリカ人との間の類似性を見るのは容易である。そのような人々には、生前はどこか周辺的だったが、彼らの時代に関係する高い歴史的重要性を獲得していった人々、超 絶主義哲学者で『市民的不服従』の著者のヘンリー・デイヴィッド・ソロー（Henry David Thoreau）、シエラクラブの創設者で現代の環境運動の創始者のジョン・ミューア（John Muir）がいる。また、ウィリアム・ジェニングス・ブライアン（William Jennings Bryan）、別名、「偉大な一般人（The Great Commoner）」のような人との類似性を見るのも容易である。彼はポピュリスト運動の指導者で独占の敵で強力だった。しかし、歴史的に重要でない人物 として霞んでいった。

ストールマンは、ソフトウェアを公共的財産と考えた最初の人間ではないが、GPL のおかげで未来の歴史書の脚注は保証されている。この事実を前提に一歩下がり、現在の時間枠の外からリチャード・ストールマンの遺産を検証することは価値 があるように思える。GPL は、2102 年にもまだソフトウェア・プログラマが使う何かだろうか、それとも、道端に捨てられているだろうか？「自由なソフトウェア」という言葉は、今日「自由な 銀」というのと同じように政治的に古めかしいものになっているのだろうか、それとも、後世の政治的出来事を恐ろしいほど予見していたのだろうか？

未来の予測は、危険な競技だ。100 年後に人々が考えているのは何かと言うのは、我々がそれに影響を持たないことを仮定しているのだと言って、ストールマンは予測を辞退する。彼が選ぶ問い は、「未来をより良くするには我々は何をするべきなのか？」だ。でも、ほとんどの人は、その問題を提示されると食いつきたくなるようだ。

「今から 100 年後、リチャードと他二名が脚注以上のものになるでしょう、」とモグレンは言う。「彼らは、話の本文と目されることになる。」

モグレンが未来の教科書の章にノミネートする「他二名」には、ストールマンの GPL の助言者で、後に電子フロンティア財団の設立者になるジョン・ギルモアと、1982 年の『リテラリーマシン』の著者のテオドール・ホルム・ネルソン（Theodor Holm Nelson）、すなわち、テッド・ネルソン（Ted Nelson）が含まれている。モグレンによれば、ストールマン、ネルソン、ギルモアの歴史的意義は際だっているが、互いに重ならない。ネルソンはふつう「ハイパーテキスト」という用語を作った人として考えられているが、モグレンは、デジタル時代の情報所有の厳しい状況を解明した点にネルソンの功績を認め る。他方、ギルモアとストールマンは、情報コントロールの否定的な政治的影響を指摘し、その影響に対抗する組織 -- ギルモアの場合は電子フロンティア財団、ストールマンの場合は自由ソフトウェア財団 -- を設立したことに大きな功績を認めている。しかし、モグレンは二人と比較して、ストールマンの活動をより個人的でより政治的でないものとして見ている。

「自由でないソフトウェアから引き起こされるであろう結果の倫理的意味が、早い時期から、彼にとって非常にはっきりしていたという点で、リチャード はユニークでした。」とモグレンは言う。「これはリチャードの人柄と大いに関係があるんですが、たいていの人は彼のことを書くときにその人柄をリチャー ド・ストールマンのライフワークの付随現象か、どうかすると欠点としてさえ描くのです。」

ギルモアは、気まぐれなネルソンと短気なストールマンの間に自分が含まれていることを、何か「あいまいな名誉」だと表現するが、モグレンの議論を支持している。ギルモアは書いている。

ストールマンの著作は、トマス・ジェファーソンのそれと同じくらい長持ちするだろうと私は思うのです。彼は、とても明解な書き手だし、自分の原理原則についても明解だ。ストールマンがジェファーソン並の影響力を持つことになるかどうか は、我々が「ソフトウェア」あるいは「技術的制約」と呼ぶ抽象的概念より、「人権」と呼ぶ抽象的概念が、今から 100 年後に、より重要なものとなりえているかどうかにかかっています。

ストールマンの遺産で見逃せないもう一つの要素は、GNU プロジェクトが開拓した協働的なソフトウェア開発モデルだ、とギルモアは書いている。たまに欠陥もあるが、それにもかかわらず、このモデルはソフトウェア 開発産業の一つの標準に進化した。最終的に、とギルモアは言う、この協働的ソフトウェア開発モデルは、GNU プロジェクトや GPL ライセンスやストールマンの開発したどの特定のソフトウェア・プログラムよりも、ずっと影響力を持つかもしれない。

インターネット以前には、ソフトウェアについて遠隔地の人が協働することは、互いに知っていて信頼しあっているチームにおいてさえ、非常に難しかったのです。リチャードはとりわけ、互いにめったに会うことがなく、組織化されていないボラ ンティアによる、ソフトウェアの協働開発の先駆者になりました。リチャードは、これを行うための基本的なツール（TCP プロトコル、メーリングリスト、diff と patch、tar ファイル、RCS や CVS やリモート CVS）は一つも作りませんでしたが、入手できたツールを使って、効率的に協

働できるプログラマの社会的集団を形成したのです。

ストールマンは、それは肯定的な評価でも、ポイントを外していると考える。「それは自由よりも開発方法を強調しているのですが、それは自由ソフト ウェアの価値観よりもむしろオープンソースの価値観の反映です。それがもしも未来のユーザーがどのように GNU プロジェクトを振り返るかという話なら、それが、開発者がユーザーを奴隷化したまま、時々ご褒美として、ユーザーを開発に役立てるが絶対に鎖を外さないよ うな世界に導くことを危惧するんです。」

スタンフォード大学の法学教授で、2001 年の『コモンズ』(The Future of Ideas) の著者のローレンス・レッシグ (Lawrence Lessig) も、同じように明るい見通しである。多くの法学者と同様に、レッシグは GPL を、昨今のいわゆる「デジタル・コモンズ」のための主要な防壁と見ている。「デジタル・コモンズ」とは、この 30 年で、インターネットを指数関数的に成長させる引き金となった、コミュニティ所有のソフトウェア・プログラムやネットワークや通信の標準の膨大な集積のこ とだ。レッシグは、人々にコンピュータ技術をより広いスケールで見るように説得したインターネットの開拓者、ヴァネヴァー・ブッシュ、ビント・サーフ、 J.C.R. リックライダー のような人たちと関連づけるよりも、ストールマンの影響力をより個人的、内省的で、つまるところユニークなものと見ている。

[ストールマンは] 議論をどうであるか、からどうあるべきか、というものに変えまし た。彼はどれだけの危機に瀕しているかを人々に自覚させ、こういった理想を推し進めるための仕組みを作った . . . 。とはいえ、サーフやリックライダーの文脈に彼をどう位置づければよいかはまったく分かりません。イノベーションが違っているのです。それはある種のコー ドに関してとか、インターネットを可能にしたということではない。[それは] むしろ、インターネットが持つある種の価値に人々を開眼させたことにありま す。後にも先にも、このような人は他にいないと思います。

もちろん、誰もが、ストールマンの遺産を石に刻まれたものと見ているわけではない。エリック・レイモンドは、オープンソースの主導者で、1996 年以降、ストールマンのリーダー的役割がかなり衰退してきていると感じており、2102 年の水晶玉を覗いたとき、そこにあいまいなシグナルを見る。

ぼくは、ストールマンの作品 (GPL、Emacs、GCC) は、革命的作品として、情報世界の礎石として見られるだろうと思う。歴史は、RMS がそれにもとづいて行動したいくつかの理論には、それほど親切でなく、縄張り根性的、カルトリーダー的にふるまう彼の個人的傾向には、まったく親切でない だろうと思う。

ストールマン自身も、あいまいなシグナルを見る。

今から 20 年後、GNU プロジェクトを歴史が何と言うかは、公共の知識を使う自由のための戦いに誰が勝つかにかかっているでしょう。ぼくらが負ければ、ぼくらは脚注になるだけで しょう。ぼくらが勝っても、人々がそのシステムを「リナックス」だと思っていれば、GNU オペレーティング・システムの役割が知られているかどうか、それは定かではありま

せん。人々は何が、なぜ、起こったかについて間違った像を描くでしょう。 でも、ぼくらが勝ったとしても、今から 100 年後の人たちがどういう歴史を学ぶかは、そのとき誰が政治的に支配しているかによるでしょうね。

19 世紀に歴史的例示を求めて、ストールマンは、ジョン・ブラウンという人物を呼び出す。彼は奴隷廃止論の闘士で、メイソン・ディクソン線（アメリカ南部と北部を分ける州境の線 -- 訳者）の片側では英雄視され、反対側では狂人視されていた。

ジョン・ブラウンの奴隷反乱は全く不成功だったが、それに続く裁判の過程で、彼は奴隷制廃止を求める国民的な要求を効果的に喚起した。内戦（南北戦 争）中は、ジョン・ブラウンは英雄だった。100 年後、1900 年代のほとんどの間、歴史の教科書は、彼は狂っていたと教えた。合法的人種隔離の時代、偏見は恥知らずなことであったにもかかわらず、アメリカは、南部が 自分たちについて語りたい物語を一部受け入れ、歴史の教科書は、内戦と関連する出来事について多くの嘘をついた。

そのような比較は、現在の仕事の周辺的性質にストールマン自身が気付いていることと、彼の評価が現在二分されていることの双方を示している。ストー ルマンの評判が、再建期後のブラウンの汚名並に貶められるようなことはありそうにない。時折、戦争のような例えを持ち出すが、それにもかかわらず、ストー ルマンが暴力をインスパイアすることはまずない。しかし、ストールマンの理想が灰の山で終わる未来を思い描くことはたやすい。4

しかしそう思っていると、この意志こそが、いつかストールマンの最大で永遠の遺産だった、ということになるのかもしれない。この十年間の間近からの 観察者、モグレンは、ストールマンの生涯の「業績」に対して、ストールマンの人柄を、非生産的とか、付随物と見誤る人に注意を喚起している。その人柄抜き では、話題にすべきものはほとんど生まれなかっただろう、とモグレンは言う。最高裁判所の書記をしたモグレンは言う。

いいかな、ぼくが仕えた最も偉大な男は、サーグッド・マーシャル（Thurgood Marshall）だった。ぼくは何が彼を偉大な人間にしたのか、なぜ、彼に可能なやり方で世界を変えることができたのかを知っている。二人を比較するの は木の枝先まで行くようなもので、少し危険だ。二人は一致するところがあまりないからね。サーグッド・マーシャルは、社会の中の人だった。マーシャルは、 マーシャルたちを閉め出した社会から見捨られている社会を代表していた。しかし、それでも社会の中の人だった。彼のスキルは社会的スキルだった。しかしま た、彼はすべてに一貫した人だった。他のすべての点で彼らが違っていたとしても、その意味で、彼と比較できる人間といえば、-- 始めから終わりまで一貫していて、コンパクトで、星々をつくる実質でできている -- それはストールマンだよ。

そのイメージを納得させようとして、モグレンは、2000 年の春に共有した瞬間の話をした。VA リナックスの新規株式公開の成功がまだビジネスメディアにこだましていて、自由ソフトウェアに関する半ダースほどの話題がニュースの中を泳いでいた。コメ ントを求められる論点と記事の渦巻く嵐に囲まれながら、ストールマンとランチの席についたときの台風の目に入った遭難者のような気分をモグレンは思い出 す。次の一時間、会話は静かに

ある一つの話題を巡っていた、と彼は言う。GPL を強化するという話題だ。
「ぼくらはそこに座って、東欧のいくつかの問題とどう取り組むのか、コンテンツの所有という問題が自由ソフトウェアの脅威になり始めたらどうするの か、を話し合っていた。」モグレンは回想する。「話をしているときに、通りすがりの人から、どういうふうに見えるのだろうということが頭をかすめた。ここ に二人、ちょっとひげづらのアナーキストがいて、次の一歩を構想し、計画している。そして、もちろんリチャードは髪のほつれをちぎってはスープに落してい る。いつものことなんだ。ぼくらの会話を聴いている人はみんな、ぼくらをクレイジーだと思ったことだろう。でも、ぼくは知っていた。いままさにこのテーブ ルで革命は起こっている。これが革命を起こすんだ。そして、この男がそれを起こすんだ、とね。」
モグレンは言う、あのときほど、ストールマンのスタイルの根源的なシンプルさを実感したことはなかった。
「ファニーだったねえ。」モグレンは回想する。「彼に言ったんだ。『リチャード、きみとぼくは、この革命でさっぱり金儲けをしなかったね。』そして、ランチはぼくが払った。彼はランチに払う金を持ってないことを知っていたからね。」5
後注
1 サンは、商標の異議申し立てにより、ぎこちない名前の「オープンオフィス・ドット・オルグ（OpenOffice?.org）」を使うことを余儀なくされた。
2 マルコ・ボエリーズ、著者とのインタビュー（2000 年 7 月）
3 ジマーマンの法的苦闘の追加情報は、スティーブン・レヴィの『暗号化』を読まれたい。本書第一版の最初の書籍版で、私は、モグレンは国家安全保障局に対す るジマーマンの防御を助力していたと書いた。レヴィの説明によれば、ジマーマンは、国家安全保障局（NSA）ではなく、米連邦地検と米関税局に捜査されて いた。
4 RMS: サム・ウィリアムズが[第一版で - 訳者]さらにここで言っていた言葉、「自由ソフトウェアという理想を、大衆運動ではなくて、占有的な誘惑の力に対する一連の個人的戦いという形にするとき に」というのは事実に合わない。GNU プロジェクトの最初の発表のとき以来、私はその目標の支持を公衆に求めてきた。自由ソフトウェア運動は大衆運動であることを目指していて、唯一の問題はそ れが「大衆（mass）」と言うのに充分な支持者を持っているかということだ。2009 年現在、自由ソフトウェア財団には、高額の会費を支払う約三千人の会員と二万人以上の電子メールの月刊ニュースレター購読者がいる。
5 RMS: 私は人々が御馳走してくれるときに決して断らない。なぜなら、私のプライドは御勘定を持つところにはないからだ。しかし、当然、ランチに支払うお金はちゃ んと持っていた。私の収入は、約半分が講演収入で法学教授のサラリーにかなわないけれども、貧乏ではない。

サム・ウィリアムズからのエピローグ
孤独を打ち破る

［RMS: この章は、サム・ウィリアムズからの非常に個人的なものなので、全ての変更を角括弧または省略符合で示した。そして、技術的なポイントまたは法的なポイントを明確化する場合に限り変更し、敵意や情報の欠如に見えたものに限って文章を削った。また、幾つかの点に答えるために`RMS:'のラベルをつけて注釈 した。ウィリアムズもこの章の本文を変更しているが、それらは明示されていない。］

生きている人物の評伝を書くことは、芝居の上演にいくらか似たところがある。幕前で演じられるドラマは舞台裏のドラマよりしばしば見劣りする。

アレックス・ヘイリーはマルコム X の自伝の中で、この滅多に見られない舞台裏のドラマを見せてくれた。ヘイリーはゴーストライターとしての役割を一歩踏み出し、本のエピローグに自分の声を 込めた。このエピローグは、はじめはネイション・オブ・イスラムのスポークスマンから、権力の道具かスパイとして追い払われていたフリーランスの記者が、いかにして人的・政治的な防壁をかいくぐって、マルコム X の人生の物語を書き下すに至ったかを説明している。

この本をマルコム X の自伝と比較することはいささか躊躇するが、あの腹蔵のないエピローグに対して、私はヘイリーに感謝している。それはこの一年以上に渡って、その全キャリ アを気難し屋として築いてきた伝記上の人物の扱い方に関する、一種の取扱説明書の役割を務めてくれた。［RMS: 私は、他人が大きな疑問を持たないで受け容れていることにノーと言ってキャリアを築いてきたが、もしも私が時に気難しく見えるか、または気難しいとして も、そこに特別の意図はない。］書き始めたときから、私は同じようなエピローグでこの伝記を終えたいと思った。それはヘイリーへの表敬であると同時に、この本の来歴を読者に知ってもらう 一つの方法としてである。

この本の背景について物語は、オークランドのアパートに始まり、本書で言及したさまざまな舞台--シリコンバレー、マウイ、ボストン、ケンブリッジ --を巡っていく。だが、結局のところ、それは、世界の出版の中心地、ニューヨーク州ニューヨークと、ソノマ郡の出版の中心地、カリフォルニア州セバスト ポルという二つの街の物語だ。

話は 2000 年 4 月に始まる。そのとき私は、ビーオープン（BeOpen?）という不運なウェブサイト（http://www.beopen.com/） のために記事を書いていた。私の最初の仕事は、リチャード・ストールマンと電話インタビューをすることだった。インタビューは成功し、VA ソフトウェア（旧 VA リナックス・システム、その前身は VA リサーチ）が所有する人気の「おたく系ニュース」サイト、スラッシュドット（http://www.slashdot.org/）の特別記事の日替りリストにリンクされた。数時間のうちに、ビーオープンのウェブサーバーはクリックでサイトにやってくる読者で熱くなっていった。

どう見ても、話はそこで終わるはずだった。だが、インタビューの三ヶ月後、カリフォルニア州モンテレーで行なわれたオライリー・オープンソース・カ ンファレンスに出席しているとき、私はニューヨークの大手出版社で国外権利の管理をしているトレーシー・パティソンから次のメールを受け取った。

宛先：sam@BeOpen.com
件名：インタビュー
日付：2000 年 7 月 10 日 月曜日 15:56:37 -0400

拝啓　ウィリアムズ様

ビーオープンのリチャード・ストールマンとのインタビューをとても興味深く読みました。私は以前から RMS とその仕事に関心をもっており、記事を見つけたときはうれしく思いました。記事は、GNU/Linux と自由ソフトウェア財団によってストールマンが実現しようとしていることの精神をとらえた、実に素晴らしい仕事だと思いました。

しかし、もっと続きが読みたいのです。それは私一人ではないと思います。あなたのイ ンタビューにもっと広がりを持たせ、最新のものにするような情報やリソースがもっとあり、それで、ストールマンのプロフィールをより膨らませていけると思 いませんか。たぶん、彼の人柄や経歴に関する逸話的な情報をもう少し盛りこむと、コアなプログラミングの現場の外にいる読者にとっても、実に面白く啓発的 なものになると思います。

メールには、その提案についてもっと話がしたいので電話をして欲しいと書いてあった。私はすぐにそうした。トレーシーは彼女の会社が電子書籍の新シ リーズを立ち上げようとしており、新規読者にアピールするような物語を欲しがっていると話した。電子書籍の形式は三万語で約百ページ、彼女はハッカーコ ミュニティの主要人物のプロフィールを描くというアイデアを上司に売り込んでいた。上司はそのアイデアが気に入った。そして、そのための興味深い人物を探 しているときに、ビーオープンの私のストールマンとのインタビューに行き当たり、私にメールが来たというわけだ。

あれは、トレーシーがインタビューを長編の人物紹介に拡大する気はないかと聞いたときだった。

ありますとも、と私は即答した。トレーシーは、それを受け入れる前段として、彼女の上司に見せられるような記事の提案書をまとめることを提案した。 二日後、よく練った提案書をまとめて彼女へ送った。一週間後、トレーシーから結果のメールが来た。ボスがゴーサインを出したのだ。

ストールマンに電子書籍プロジェクトへ参加してもらうことは、こちら側の一方的な考えだったことを認めるほかない。オープンソースの鼓動を追ってき た記者として、私はストールマンがうるさ型だということを知っていた。私はすでに、"GNU/Linux" と言わずに "Linux" を使っていると指摘したお叱りメールを半ダースほど頂戴していた。

しかし私は、ストールマンが社会の一般の人々に彼のメッセージを伝える手段を探していることも知っていた。多分、彼にこのプロジェクトをそういうふ うに提示したら、受け

入れてくれる可能性は増すだろう。そうでなくても、私はストールマンがインターネットのあちこちに残している豊富な文書、インタ ビュー、オンラインの会話の記録に常時頼ることができたから、本人未承認の伝記を執筆することもできた。

調査をしていると、2000 年 6 月号の MIT テクノロジーレビューにストールマンが書いた「自由か著作権か?」というエッセイにぶつかった。エッセイは、電子書籍をソフトウェア的罪悪の詰め合わせだ として酷評していた。読者が電子書籍を読むためには、占有的ソフトウェアプログラムを使う必要があるばかりか、無許可コピーを防ぐために過度に厳しい手法 を使っているとストールマンは嘆いていた。譲渡可能な HTML や PDF ファイルではなく、読者は暗号化されたファイルをダウンロードしなければならなかった。つまるところ、電子書籍の購入とは、暗号化された内容を解読するた めの譲渡できないキーの購入を意味していた。認証されたキーを使わずに本の内容を開こうとする試みは、いかなるものでもデジタル・ミレニアム著作権法違反 という犯罪を構成することになる。1998 年の法律は、インターネットで著作権が通用するように支援するという構想だった。本の内容をオープンなファイル形式に変換した読者にも同様な刑罰が用意さ れていた。自宅の別のコンピュータでその本を読もうとしたにすぎないときでも、そうなのである。通常の本とは違って、読者はもはや、電子書籍を貸したり、 コピーしたり、再び売ったりする権利を持っていなかった。読者は許されたマシンでそれを読む権利しかないのだ。ストールマンは警告した。

紙の本を使うことに関しては、私たちにはまだ昔と変わらぬ自由があります。だが、電 子書籍が印刷された書籍に取って代ったら、この例外はなんの役にも立たなくなるでしょう。新しいテキストを、一見印刷された紙のように見える媒体上にダウ ンロードすることを可能とする「電子インク」によって、新聞ですら一日限りのものになりかねません。想像してみて下さい。古本屋はなくなり、友達に本を貸 すこともなくなる。公共の図書館から本を借りることもなくなり、金を払わずに人に読む機会を与えるような「漏洩」もなくなる。(また、マイクロソフト・ リーダーの宣伝文句から判断するなら、名乗らずに本を購入することもできなくなる。) これが、出版業者が私たちのために思い描いている世界なのです。1

エッセイを読んで心配になったのは言うまでもない。トレーシーも私も、彼女の会社が使おうとしているソフトウェアについて議論していなかったし、電 子書籍の使用を支配することになる著作権の[ライセンスの]タイプについても議論していなかった。私はテクノロジーレビューの記事に触れて、彼女の会社の 電子書籍の方針について情報をもらえないだろうかと頼んだ。トレーシーは折り返し連絡すると約束してくれた。

早くとりかかりたかったので、ともかくストールマンに電話をして本のアイデアを伝えることに決めた。電話すると、ストールマンはすぐに興味と懸念を示した。「ぼくが電子書籍について書いたエッセイは読みました?」と彼は尋ねた。

「ええ、そのエッセイを読んで出版者の返事を待ってるところです。」と言うと、ストールマンは二つ条件をつけた。自分がそもそも反対している電子書 籍のライセンスの仕組み

に手を貸したくないし、結果として手を貸すようなこともしたくない。「偽善者のように見えることにはどんなことにも参加したくない んだ。」と彼は言った。

ストールマンにとって、ソフトウェアの問題は著作権の問題に較べれば二の次だった。ストールマンは、会社が著作権の中で、読者が電子書籍の内容のそ のままコピーして配布するのは自由だと明言するなら、出版者やサードパーティーベンダーがソフトウェアに何を使っても無視しようと言った。彼は、ありえる モデルとして、スティーブン・キングの『プラント』をひいた。2000 年 6 月、キングは彼の公式ウェブサイトで『プラント』を連載物として自主出版すると公言していた。本の値段は全部で 13 ドル、連載の一回分は 1 ドルだった。キングは、読者の少なくとも 75 パーセントがそれぞれの章に支払ってくれれば、新しい章のリリースを続けると約束した。8 月までは、プランはうまくいっているようであり、キングは最初の二つの章を出版し、第三章にとりかかっていた。

「こういうのなら、受け入れられるよ。」とストールマンは言った。「そのままのコピーも許していればね。」[RMS: 思い出してみると、私は暗号化についても問題提起をしていた。続く二つ目のパラグラフはそれを裏付けている。私は、読もうとするときに自由ではないプログ ラムを要求するいかなる本の出版にも同意しなかっただろう。]

私はトレーシーにこの情報を伝えた。彼女と私は、公正な取り決めができそうなことが確信できたので、ストールマンに電話して、本の最初のインタ ビューの段取りをつけた。ストールマンは権利関係についてはそれ以上詮索せずにインタビューに応じた。最初のインタビューがすむと、すぐに次のインタ ビューの予約を取りつけて、ストールマンがタヒチに二週間のバケーションに行こうとする前にそれを実現した（キヘイでのインタビューがそれだ)。[RMS: それは純粋な休暇ではなかった。そこで私は講演もしたのだった。]

ストールマンの休暇中にトレーシーから悪い知らせが届いた。彼女の会社の法務部は電子書籍の著作権[のライセンスの]表示をいじりたがらなかった。 自分の本を譲渡可能にしたい読者は[まず、暗号コードを破る必要があり、それを HTML のようなオープンなフォーマットに変換する必要があった。これは非合法化されているため、刑罰に見舞われることになるかもしれなかった。]

二つの最新インタビューが手中にあり、この真新しい材料抜きで書くなんて考えられなかった。私はすぐニューヨーク便を手配して、エージェントとトレーシーに会って妥協案を探ろうとした。

ニューヨークに飛んだ私は、私のエージェントであるヘニング・ガットマン（Henning Guttman）に会った。顔をあわせるのは初めてだった。ヘニングは、少なくとも、出版社側に妥協案を押し込むチャンスについては悲観的だった。既成出 版社の大勢は、電子書籍形式をすでに十分疑わしいものと見ていたので、読者が支払いを回避しやすくなる著作権文言を実験するムードではなかった。しかし、 ヘニングは技術書専門のエージェントとして、私の苦境の新奇な性格に興味をそそられた。私は、すでに集めていた二つのインタビューと「偽善者のように見え る」形では本を出版しないという約束について話した。私が道義的

に拘束されていることにヘニングは同意して、そこを交渉のポイントにすることを提案した。

それがなければ、私達はいつでもアメとムチのアプローチを取れたね、とヘニングは言った。アメは、ハッカーコミュニティ内部の倫理を称賛する電子書 籍を出版することによるパブリシティーだ。ムチは、ハッカー倫理にもとる電子書籍の出版に伴うリスクだ。私達は、ドミトリー・スクリャーロフがインター ネットの大義の象徴になる 9 ヶ月前に、気鋭のプログラマが電子書籍をハックする方法を公開するのは時間の問題だということを知っていた。また、ストールマンの[暗号化した]電子書籍 を大出版社がリリースすれば、それは表紙に、ソフトウェア的に「この電子書籍を盗んでごらん」と書くようなものだということも。

ヘニングと会った後、ストールマンに電話した。アメをもっと魅力的にできないものかと願いつつ、私は可能な妥協案についていくつか議論した。本の内 容を出版社が[二重]ライセンスでリリースする、というのはどうだろう。自由ソフトウェアの文房具ソフトウェア集であるオープンオフィスでサン・マイクロ システムズが採用したように。 出版社は電子書籍の DRM で制限された[その通常の]形式のヴァージョンをリリースすることができて、2 電子書籍につきもののあらゆる飾りものを利用できる一方で、美しさには劣るが、喜ばしい HTML 版をリリースするのだ。

ストールマンは私に、[二重のライセンスの]考え方はかまわないが、自由にコピー可能な版を制限付きのヴァージョンより劣ったものにするような考え 方はおぞましいと告げた。それに[二番目の考えについて、今回は、そもそも違っている。なぜなら自分には]結果をコントロールする方法がある、とストール マンは言った。協力を断ればよいのだ。

[RMS: 問題は、制限されたヴァージョンに私が同意するのが正しいのかどうか、ということだった。私は、サンのオープンオフィスの自由ヴァージョンを支持できる、 なぜなら、それは自由ソフトウェアだし何もないよりずっと良いからだ。他方、同時に、自由ではないヴァージョンはお断りである。ここに自己矛盾はない。サ ンにとっては、自由ではないヴァージョンを出すのに私の了解を得る必要はないからだ。その存在に私の責任はない。この場合には、自由にコピーできない ヴァージョンを私が了解すれば、その責任は私が負わなければならなかっただろう。]

私はさらに幾つかの提案をしたが無駄だった。ストールマンから引き出せた電子書籍に関する唯一のライセンスの制限は、ファイル共有形態への制限は「非商用的再配布」に限る、という譲歩 [RMS: つまり、妥協の追加] だけだった。

交渉を終える前にストールマンは私に、私がこの著作を[自由に共有可能なもの]にすることをストールマンと約束したと、出版社に話すように提案して きた。その発言には同意できないが、[RMS: それでも、彼が私の条件を最初に受け入れたときから、それは真実だったのだ。] あなたの協力がなければこの本は完成できないと思っている、と私は言った。ストールマンは一応満足したように思われ、いつもの締めくくりの言葉、「ハッ ピー・ハッキング」でお開きになった。

翌日、ヘニングと私はトレーシーと会った。トレーシーは、会社は暗号化しない形式でコピー可能な抜粋を出版するつもりはあるが、抜粋は 500 語以内に制限することになると言った。ヘニングは、それでは私がストールマンに対する倫理的責任を果たすのに不十分であることを彼女に説明した。トレー シーは、例えば、アマゾン・ドット・コムのようなオンラインベンダーに対する自社の契約上の義務について話した。会社が今回に限り電子書籍の内容を公開す ると決めたとしても、パートナーに契約違反呼ばわりされる危険がある、というのだ。経営陣もストールマンも心変わりしないのなら、私が決めるしかなかっ た。インタビューを使い、ストールマンとの以前の合意を破るのか、記者倫理を守り、この本を書く口約束を反故にするのか。

この会見の後、エージェントと私は、場所を変えて三番街のパブに行った。彼の携帯電話を使ってストールマンに電話し、留守番電話にメッセージを残した。ヘニングはしばらく席を外して、私が考えをまとめるための時間をくれた。戻ってきたとき、彼は携帯電話を掲げていた。

「ストールマンだ。」とヘニングは言った。

最初から気まずい会話になった。私は、出版社の契約上の義務というトレーシーの説明をそのまま伝えた。

「それで、」とぶっきらぼうにストールマンは言った。「何で会社の契約上の義務をぼくが気にしなきゃいけないのかな。」

大出版社に対して、三万語の電子書籍のことで、ベンダーと法廷で争うような危険を冒すように求めるのは法外な注文だ、と私は示唆した。

[RMS: 彼の暗黙の前提は、ただの原則のためには、私はおそらくこの取引を断れないだろうということだった。]

「わからないのかい。」とストールマンは言った。「ぼくがこうしているのは、まさにそのためなんだ。ぼくは合図になる勝利が欲しいんだ。彼らにいつもの商売か自由かを選んで欲しいんだよ。」

「合図になる勝利」という言葉が私の頭にこだましている間、しばし私の目は舗道を行きかう人の足元をさまよった。バーに戻った私は、嬉しいことにこ の場所がミュージシャンだった頃に私がよく演奏していた 1976 年のラモーンズの歌、『53rd and 3rd』の街角から半ブロックも離れていないことに気づいた。その歌に描かれた年中挫折している街頭のハスラーのように、まとまったと思ったとたんに話が こわれていくように感じられた。皮肉は明白だった。数週間に渡って、上機嫌で他人の嘆きを記録した後で、自分がリチャード・ストールマンの譲歩というまず ありえない手柄を引き出そうとしている立場に立っていた。私がずっと言葉を濁して、出版社の立場を擁護し、より大きな同情を表明すると、ストールマンは、 血の匂いをかぎつけた獣のように攻撃してきた。

「そういうわけなんだ。ぼくからしぼり取ろうというの ? きみは彼らの意思に屈するところなんだね。」

［RMS: 引用文は、ウィリアムズのこの会話の受け止め方が全く間違っていることを示している。彼は私を捕食者と比較しているが、私は受け入れ困難な取引を迫られ て、それは駄目と言ったに過ぎない。すでに、私は幾つかの歩み寄りをしていた。私は自分の原則とおさらばするのを断っただけだった。私は、しばしば譲歩す る。満足しない人は、私が「全然譲歩しない」と言うが、誇張だ。http://www.gnu.org/philosophy/compromise.html を参照されたい。当時、私は、彼が同意していた条件を無視して、私の拒絶にもかかわらず、DRM付きの本を出版しようとしているのではないかと危惧していた。私が嗅ぎ付けたのは彼の「血」ではなく裏切りの可能性だった。］

私はまた二重著作権の問題を持ち出した。

「ライセンスのことだね。」とストールマンはそっけなく言った。

「そう、ライセンスのことです。著作権じゃなく」と言いながら、突然、水中で血の噴流をつくっている傷ついたマグロになったように感じた。

「ああ、なんで僕の言った通りにしなかったんだ。("Aw, why didn't you just fucking do what I told you to do!")」と彼は吠えた。[RMS: 副詞として"fucking"を使っているのは、決して私のスピーチパターンの一部ではないこと、状況に適合していないこと、の両方から、この引用文には 改竄があったと思う。彼の引用する言葉は、反抗的な部下への叱責だ。私は彼には倫理的義務があると感じていたが、彼は部下ではないし、そのように話しかけ たりしない。レコーダーではなくてノートを使ったために、彼は信頼性のある正確な言葉を残せなかったのだ。］

私はメモに、「まあいいか。彼らのやってることは邪悪だよ。ぼくはそれを支持できないんだ。じゃあね。」というストールマンの最後の古臭いセリフを 記しているから、最後まで出版社の立場で議論を続けていたに違いない。［RMS: 彼は聞く耳を持たず、彼の立場を受け入れずに会話を終わらせるには、一方的に電話を切るしかない、と私は結論付けたようだ。］電話を置くと、エージェントが、注ぎたてのギネスを滑べらせて寄こした。「これが必要かと思ってね。」と彼は笑って言った。「最後まで動揺していたよう だったが。」確かに動揺していた。動揺はギネスを半分以上飲むまで収まらなかった。自分が「邪悪」の使者になぞらえられるの聞くのは変な気分だった。［RMS: 引用されている通り、私の言葉は出版社を批判しているが、ウィリアムズ個人を批判しているのではない。もし彼がそれを個人的に受け取ったのならば、たぶ ん、彼が私に受け入れるように迫った取引に倫理的責任を取り始めたことを示唆している。］次の記事のアイデアをひねり出そうとしてオークランドのアパートに座っていた三ヶ月前のことを思うと、さらに変な気分だった。今は、ロックの歌詞を通じて 知っていたにすぎない世界の一部に座っている。出版社の役員と会合を持ち、エージェントとビールを飲み、前日までは決して見ることもなかったものを見てい た。全てがあまりにもシュールだった。自分の人生を映画の合成技術を使って振り返っているようだった。

その時、私の中のバカ度計測装置にスイッチが入った。最初の動揺は笑いの発作に道を譲

った。エージェントには、私が場違いに感情を乱した、傷つきや すい作家の一人に見えたことだろう。私はといえば、自分の状況の逆説的な素晴らしさが分かりかけていた。売れようが売れまいが、かなり良い記事ができあ がっていた。問題はそれをどこに載せるかということだけじゃないか。やがて笑いの発作が収まると、私はコップを持って乾杯のしぐさをした。

「友よ、最前線へようこそ。」と言って、エージェントのジョッキにかちんとあわせた。「楽しもうじゃありませんか。」

この話がお芝居だったら、ここでちょっとロマンチックな幕間劇が入るところだろう。私達の会合がピリピリしていたことに気落ちしたトレーシーが、同 僚とお酒を飲みに行こうと私とヘニングを誘ってくれた。私達は三番街のバーを後にしてイーストビレッジに向かい、トレーシーとその同僚たちと落ち合った。

そこでは仕事の話を注意深く避けてトレーシーとおしゃべりをした。会話は楽しくてリラックスしたものだった。別れ際に翌日の夜も会う約束をした。このときもまた会話は愉しく、あまりに愉しかったので、ストールマンの電子書籍のことは遠い昔の思い出になっていた。

オークランドに帰って、ジャーナリストの友人知人にずいぶんたくさん電話をした。私は自分の苦境を語った。インタビュー前の交渉でストールマンに多 くを与えすぎた点を叱る意見が多かった。[RMS: 本を全部読んだ人は、私が条件を断念しないことが分かっただろう。] 元ジャーナリズム学部の教授は「偽善者」についてのストールマンのコメントを無視して記事を書けばいい、と示唆した。ストールマンのメディアへの精通ぶり を知っている記者は同情を示したが、きみが決めることだ、という反応はみな同じだった。

私は本を後回しにすることにした。インタビューはあったが、あまり捗っていなかったのだ。またそうすることで、まずヘニングを介さずに、トレーシー と話をする機会が与えられた。クリスマスまでには、私達はもうお互いに行き来するようになっていた。彼女がウエストコーストに一度飛来し、私がニューヨー クに二回飛んだ。大晦日の前日、私はプロポーズした。どちらの海岸に住むかを決めるに当たって、私はニューヨークを選んだ。二月までに、ラップトップ・コ ンピュータとストールマンの伝記に関する全調査ノートを荷造りし、私達は JFK 空港に向かって飛び立った。トレーシーと私は五月十一日に結婚した。失敗した本との引替えにしては出来すぎだった。

夏の間に、インタビューノートを雑誌記事にしては、と考え始めていた。倫理的にも、そうして問題ないと感じた。もともとのインタビューの条件に、伝 統的な印刷メディアに関することは何も入っていなかったからだ。正直なところ、八ヶ月間音信を断った後で、ストールマンについて書くのはいくらか気分のい いことだった。九月の電話での会話以来、ストールマンからはメールを二通もらっただけだった。どちらもウェブマガジンのアップサイド・トゥデイの二つの記 事で "GNU/Linux" ではなく "Linux" を使っているとのお叱りだった。それを別にすれば、私はこの沈黙を楽しんでいた。六月、ニューヨーク大学での

講演の約一週間後に、ストールマンについて５ 千語の雑誌向けの長さの記事を書いてみた。このときは言葉が流れ出て来た。距離が、失っていた感情の遠近感を回復させてくれたのだと思う。

七月、トレーシーに最初のメールをもらってからちょうど一年後に、ヘニングから電話があった。彼は、カリフォルニア州セバストポルの出版社、オライ リー・アソシエイツがストールマンの記事を伝記として出版することに関心を持っていることを告げた。［RMS: オライリーと契約することを提案したのかどうか記憶は曖昧で、今となっては確かなことを言えない。］私はそのニュースを喜んだ。エリック・レイモンドの『伽藍とバザール』を出版した会社でもあり、オライリーは、世界中のどの出版社より、初期の電子書籍を 葬り去った問題に敏感だと思われた。一人の記者として、私はオライリーの本、『オープンソース』を歴史的参考資料として重用していた。その本の多くの章 は、ストールマンの執筆した章を含め、再配布を許可する著作権[のライセンス]表示付きで出版されていることも知っていた。電子出版が問題になるとき、そ ういう知識は役に立つだろう。

はたせるかな、その問題が持ち上がった。私はヘニングを通じて、オライリーは伝記を書籍としてだけでなく、これから始めるサファリ技術書オンライン 購読サービスの一部としても出版するつもりだということを知った。サファリのユーザーライセンスには特別の制約がある、とヘニングは注意を促した、3 だがオライリーは、媒体の如何を問わず、ユーザーが本の文章をコピーして共有することを許可するような著作権にすることを認める意向だった。著作者として 私には、基本的に二つのライセンスの選択肢があった。公開出版ライセンスか GNU 自由文書ライセンスかの。

私はそれぞれのライセンスの内容と背景を子細に点検した。公開出版ライセンス（OPL; Open Publication License）4 は、複製された作品が公開出版ライセンスを保持する限り、作品の全部まはた一部を「物理的あるいは電子的な」任意のメディアで複製し配 布をする権利を読者に与える。一定の条件に従う限り、作品の修正も許される。最後に、公開出版ライセンスには多くの選択条項があり、著作者が選べば、著作 者の事前の承認のない「実質的な修正」版や、紙の本の形式の派生物の出版を制限することができる。

他方、GNU 自由文書ライセンス（GFDL; GNU Free Documentation License）は、同一のライセンスを持っている限り、任意の媒体で文書を複製し配布することを許可する。一定の条件に従う限り、文書の修正も許可す る。5 しかし、OPL と異なり、著作者にある種の修正を制限するような選択肢を与えない。GFDL は著作者に、競合する本を生み出すことになるような修正を拒否 する権利も与えない。だが、著作権者以外の当事者が GFDL に保護された作品を百部以上出版したいときは、表紙と背表紙が一定の形式を持つことを要求す る。

これらのライセンスを調査する過程で、GNU プロジェクトの「さまざまなライセンスとそれらについてのコメント」というウェブページも抜かりなく見 に行った。6 そのページで公開出版ライセンスに対するストールマンの論評を見つけた。ストールマンの批判は、修

正された作品が生まれる一方で、著作者が OPL の選択条 項のどれか一つを選んで修正を制限することができるようにしていることに関するものだった。もし著作者がどの選択条項も使いたくなければ、GFDL を使う ほうが良い、とストールマンは注意する。そうすれば、文書の修正版に突然選択条項が現われる、というリスクを最小にできるから。

どちらのライセンスでも修正を重要視するのは、ライセンスの当初の目的を反映してのことだ。つまり、ソフトウェア・マニュアルの所有者に、マニュア ルを改良し、改良点をコミュニティに公開する機会を与えることだ。私の本はマニュアルではなかったから、どちらのライセンスの修正条項もほとんど気になら なかった。私の唯一の関心は、ハードカバーの本を購入したときに楽しめるのと同じ自由、本のコピーを交換したり中身のコピーをつくる自由をユーザーに与え ることだった。どちらのライセンスでもこの目的に適っていると判断したので、私は届けられたオライリーの契約書に署名した。

とはいえ、無制限の修正という考え方が私の興味をそそった。トレーシーとの交渉を始めた頃、私は、電子書籍の内容に対して GPL スタイルのライセン スがもつメリットを説いた。最悪の場合でも、と私は言った、このライセンスはこの電子書籍に、良い意味で多くの注目を集めることを保障するだろう。うまく いけば、読者が著作のプロセスへ参加することを促すだろう。著作者の一人として、私の名前が常にトップに掲載される限り、私は他人が自分の作品を改善でき るようにしたかった。それに、本が進化していくのを見るのは、興味深いことでさえあるだろう。私は、中央の欄に私の本文があり、それを周囲の啓発的な第三 者のコメントが囲むような、オンライン版のタルムードとでもいうべき後続版を思い描いた。

私のアイデアは、もともとテッド・ネルソンが 1960 年に着想した伝説のソフトウェア概念、プロジェクト・ザナドゥ(http://www.xanadu.com/) から吹き込まれたものだ。1999 年のオライリー・オープンソース会議のとき、私は、このプロジェクトの[自由な]派生物である Udanax の最初のデモを見て、その成果にうなった。あるデモの中で、Udanax は親文書と派生作品を似たような二段組のプレーンテキスト形式で表示していた。ボタンをクリックすると、プログラムは、親文書の各センテンスと派生作品の その概念上の継承者を結ぶ線を描き加えた。電子書籍版のリチャード・ストールマンの伝記は Udanax 使用可能でなくてもいいが、そのような技術的可能性があるなら、ユーザーにそれで遊んでみる機会を与えない手はない。7

オライリーの私の編集担当者、ローリー・ペトリツキ（Laurie Petrycki）から、OPL か GFDL かを選ぶように言われたとき、私は再びその空想にふけった。私が契約にサインした 2001 年 9 月には、電子書籍は瀕死の話題だった。トレーシーの会社を含めて、多くの出版社は、利益が出ないという理由で電子書籍事業を閉鎖していた。私は思わずには いられなかった。もしも、これらの会社が電子書籍を出版形式の一つとしてではなく、コミュニティづくりの一形式として扱っていたら、これらの事業は存続し ていたのだろうか。

契約書の署名後、私はストールマンに本のプロジェクトを再開したことを知らせた。私は

また、オライリーが公開出版ライセンスか GNU 自由文書ライセンスかを選ぶことができることを話した。私は彼に、オライリーの競争相手に同じ本を表紙を替えて印刷するチャンスを与える言われはないとい う理由からだけでも、OPL がいいと思っていると話した。ストールマンは、GFDL を擁護して、オライリーはすでに何度か GFDL を使っていることを指摘した返事を書いてきた。去年色々なことがあったけれども、私は取引をしたいともちかけた。もしも、さらにインタビューする機会を与 えてくれて、オライリーが本を出版することを助けてくれるなら、私は GFDL を選ぶという取引である。ストールマンは、さらにインタビューに応じることに同意したが、宣伝関係のイベントに参加するかどうかは本の内容次第だと言っ た。これは、まずまずな条件だと思い、2001 年 12 月 17 日にケンブリッジでインタビューすることにした。

私はインタビューを、妻のトレーシーのボストン出張にあわせて設定した。出張の二日前、トレーシーはストールマンをディナーに誘うことを提案した。

「結局、」と彼女は言った。「彼が私達をひき会わせたんですもの。」

私はストールマンにメールを送り、すぐにストールマンは申し出を受けるという返信をくれた。翌日、車を運転してボストンへ行き、ホテルでトレーシー に会って、タクシーに飛び乗り、MIT に直行した。私達がテク・スクエアに到着してドアをノックすると、ストールマンはおしゃべりの真っ最中だった。

彼は「どうぞ気兼ねなく。」と言ってドアを大きく開いたので、トレーシーと私には、会話の相手の声もまる聞こえだった。それは二十代なかばくらいの若い女性で、サラと言う名前だった。

「ディナーを一緒にする人を誘いましたよ。」あのパロアルトのレストランのときと同じように、実はといった調子の猫っぽい微笑みで、ストールマンは言った。

正直に言うと、私はあまり驚いていなかった。ストールマンに新しい女友達ができたという知らせは、二、三週間前に、ストールマンの母上の厚意で私に 届いていた。「実はね、先月、リチャードが武田賞を受賞しに行ったとき、二人で日本に行ったのよ。」とそのときリップマンは教えてくれたのだ。8

レストランに着くまでの間に、サラとリチャードのなれ染めを聞き出した。面白いことにその事情は、私にとってもなじみ深いものだった。彼女は自分の 小説を書いているときに、ストールマンについて耳にし、何て興味深いキャラクターなんだろうと思った。彼女は直ちに自分の本にストールマンに基づく登場人 物をつくることに決め、登場人物の研究のためにストールマンとのインタビューを計画した。そこから状況は急に動き始めた。二人は 2001 年の年明けからつきあっている、と彼女は言った。

ストールマンの魅力を説明して、サラは言った。「個人の深刻な問題に取り組むため、政治的運動全体を築いてしまったリチャードのやり方は本当に驚きだったわ。」

私の妻がすぐ質問を投げ返した。「問題って何だったのかしら。」

「孤独を打ち破ること。」

ディナーの間、私は女性達が話すのにまかせて、この一年でストールマンがいくらかでも軟化したかについての手がかりを得ようと時間の大半を使った。 何もそれらしいものは得られなかった。以前より浮わついた感じはあったが、ストールマンには相変わらずの棘々しさがあった。あるときは、妻が強い口調で 「神様が許しません ("God forbid" とんでもないわ、の意味)」と言ったが、ストールマンの典型的なお叱りを頂戴することになった。

「悪いけど、神様なんていないんですよ。」[RMS: 私はあまりにも真面目腐っていたに違いない。私をお利口さんだったと彼が非難するのは正当だ。しかしそれは叱責ではなかった。]

やがて、ディナーがすっかり終わり、サラが帰ったとき、ストールマンはガードを少し下げたように思った。近所の本屋まで歩きながら、彼はこの一年が 人生の展望を劇的に変えたことを認めた。「ぼくは永久に一人でやって行くのかと思っていたけど、」と彼は言った。「間違っていて良かったよ。」

別れ際に、ストールマンは、最後のインタビューを段取れるように、彼の「プレジャーカード」、自分の住所、電話番号、好きな娯楽（良い本、良い食べ物、エキゾティックな音楽、ダンスなどをシェアすること）を一覧にした名刺、をくれた。

翌日、点心の食卓のむこうで、ストールマンは昨晩よりもっと恋に夢中のように見えた。カリアー寮の寮母たちとした不死の利益と不利益についての議論 を思い出しながら、いつの日か科学者が不死への鍵を用意する日がくるかもしれないという希望をストールマンは語った。「ついに幸せな人生を始めた以上、 もっと[長く人生が]続いて欲しいもんだね。」と彼は言った。

「孤独を打ち破る」というサラのコメントに私が言及したとき、ストールマンは、肉体的または精神的レベルでの孤独とハッカーのレベルでの孤独との結 びつきに気づいていなかった。「コードを共有する衝動は友情だけど、かなり低いレベルの友情だ、」と彼は言った。しかし、その後再びその話題になったと き、ストールマンは孤独、あるいは、永遠の孤独への惧れが [RMS: すなわち、ハッカーの、コミュニティのレベルの]、GNU プロジェクトの最初期の決意の火を焚きつけるのに大きな役割を演じていたことを認めた。

「ぼくがコンピュータに魅せられるのは何か他のことの結果じゃないんです。」と彼は言った。「ぼくが人気者で女性がみんなぼくに群がってきても、コ ンピュータの魅力は減りません。でも、ぼくには我が家はないと感じ、それを見つけたと思ったら失い、別の我が家を見つけたと思ったら壊されたという経験 が、ぼくに深く影響したのは確かです。ぼくが失ったものの一つは寮でした。壊されたものの一つは AI ラボでした。どんな我が家もコミュニティもないことの当て所なさはとても強烈でした。それを取り戻すためにぼくは戦おうと思ったんです。」

このインタビューの後で、私はある種、対称的な気持ちを感じずにはいられなかった。サラがストールマンの何に惹かれたのかを説明するのを聞き、ス トールマン自身が、彼を自由なフトウェアの大義に取り組ませた気持ちを語るのを聞きながら、私は、本書を書いてい

る自分自身の理由を思い出していた。 2000 年 7 月以来、私はストールマンの人柄の魅力と不快さの両方を評価することを学んだ。エベン・モグレンが以前に私に語ったように、個人を付随的現象として切り捨 てることも、自由ソフトウェア運動全体との関係を見失うことも、ひどい間違いだと感じた。色々な意味で、その二つは見分け難い形で互いに結びついているの だ。

［RMS: ウィリアムズは、肯定否定の両面から私の様々な部分について彼の反応を客観化しているが、それはまた、外見、順応、ビジネス上の成功についての彼自身の姿勢の作用でもある。］

この本を読んで、全読者がストールマンに同じレベルの親近感を持つことはないだろうが、リチャード・ストールマンほど非凡な人物像を提供できる人間 は稀にしかいないという点では多くの人が一致するはずだ。この最初のポートレートが完成したのを機に、また GFDL の助けにより、他の人が、このポートレートにその人自身の知見をつけ加えたいという、同じくらい強い衝動を感じてくれることを心から期待している。

後注

1 リチャード・ストールマン著「自由か著作権か？」（2000 年 5 月） http://www.gnu.org/philosophy/freedom-or-copyright.ja.html （翻訳：http://www.hyuki.com/gnu/frcp.html)

2 RMS: ウィリアムズは「商用」とここに書いていたが、それは誤用だ。なぜなら、それは「ビジネスと結びつけられている」こと意味するからだ。これらの全ての ヴァージョンは企業がそれらを出版するなら、商用となるだろう。[;第一版の記述は、「出版社は電子書籍につきもののあらゆる飾りものを利用した通常形式 の電子書籍の商用版をリリースできる一方で、美しさには劣るがコピー可能な HTML 版をリリースするのだ」というもの。(訳者）]

3 "Safari Tech Books Online; Subscriber Agreement: Terms of Service" を参照。 http://my.safaribooksonline.com/termsofservice 2009 年 12 月現在、これらの電子書籍は自由ではないリーダー・ソフトウェアを要求している。だから、人はそれらの使用を拒むべきだ。

4 The Open Publication License: Draft v1.0" (June 8, 1999) http://opencontent.org/openpub/ 公開出版ライセンス（非公式日本語訳） http://homepage2.nifty.com/gimpman/copyright.html を参照。

5 "The GNU Free Documentation License: Version 1.3" (November, 2008), http://www.gnu.org/copyleft/fdl.html を参照。 GNU 自由文書ライセンス（非公式日本語訳）は付録 B を参照。

6 http://www.gnu.org/philosophy/license-list.html を参照。 さまざまなライセンスとそれらについての解説の日本語は、次を参照。 http://www.gnu.org/licenses/license-list.ja.html

7 この本を Xanadu の自由ソフトウェアバージョン、Udanax に「移植」したい人は、私から熱烈な支援を受けるだろう。この好奇心をそそるプロジェクトについて、もっと知りたい人は、次のサイトへ。 http://www.udanax.com/

8 なんと、私は「人類に富と豊かさ・幸福をもたらす工学知の創造とその活用において、顕著な業績をあげた」リーナス・トーバルズと坂村健とストールマンに対 して武田計測先端知財団が授賞を決定したことを、ストールマンが賞を受け取りに日本旅行するまで、知らなかった。賞と 100 万ドルの賞金について詳しくは 財団のサイトへ。(三人の興味深いスピーチも読める-訳者_) http://www.takeda-foundation.jp/

付録　ハック、ハッカーズ、ハッキング

「ハッカー」という言葉の意味をすっかり理解したければ、長年続いているハッカーという言葉の語源論を調べることが役立つ。

ソフトウェア・プログラマのジャーゴンのオンライン版解説書、『新ハッカー辞典』は、公式的に「ハック」という言葉に含まれる含蓄を九つ列挙してい る。「ハッカー」についても九つある。しかし、同書は他方で、言葉の持つ柔軟性に騙されないように読者に警告する MIT のハッカー、フィル・アグレ（Phil Agre）を引用した付録のエッセイも収録している。「ハックには意味が一つしかない、」とアグレは論じる。「明瞭な表現を拒絶する極めて敏感で深い言葉 だ。」リチャード・ストールマンは、それを「遊び心のある賢さ」というフレーズで明瞭に表現しようと試みる。

現代のほとんどのハッカーは、定義の広狭にかかわらず、言葉の由来を MIT まで遡る。そこでは 1950 年代初めに学生のジャーゴンの人気アイテムとしてこの言葉が発酵してきた。MIT 博物館はハッキング現象を記録した特集を 1990 年に企画した。それによれば、50 年代にその言葉を使い始めた学生たちは現代の学生が「お馬鹿」（goof）という言葉を使うかもしれないところで「ハック」を使った。寮の窓におんぼろ自 動車を飾りつけるのは「ハック」だったが、不愉快で邪悪なもの、たとえばライバルの寮の窓に卵を投げつけるとかキャンパスにある肖像の外観を汚すといった ことは、そこからすべて除かれた。「ハック」の定義に暗黙のうちに含まれているのは罪の無い創造的喜びの魂だった。

この魂は、言葉の動名詞の形態に霊感を与えた、「ハッキング」だ。1950 年代の学生は、電話で午後の歓談をしたりラジオを分解することに時間の大半を費やしていたが、そんな活動は「ハッキング」と形容して良かった。これもまた 今だったら同じ活動を形容して動詞形で「馬鹿やってる」とするところだろう。

1950 年代が進むにつれて「ハック」の言葉はより鋭い反逆的な刃を獲得した。1950 年代の MIT は過度に競争的で、ハッキングは競争的文化の反動であり延長でもあった。突然、お馬鹿とおふざけは鬱憤を晴らす方法となり、キャンパスの大学管理者を軽蔑 して、学部の過酷な学生カリキュラムに窒息しそうな創造的思考と行動を満足させるものになった。大学は、鍵のかかったドアと「立ち入り禁止」看板にひるま ない学生のために、無数の通路と地下の蒸気トンネルという冒険の機会をたっぷり提供していた。学生は立ち入り禁止区域の探検を「トンネルハッキング」と呼 び始めた。地上では、キャンパスの電話システムが同様の機会を提供していた。学生は、冷静な実験と善良な注意を通じて、ユーモラスないたずらの実演方法を 学んだ。より伝統的なトンネルハッキングの追求からインスピレーションを得た学生はまもなくこの新しい活動を「電話ハッキング」と呼び始めた。

創造的な遊びと無制約な探検の結びつきの重視は、将来、ハッキングという用語が突然変異するときの基盤として貢献することになる。1960 年代の MIT のキャンパスで自らを最初にコンピュータ・ハッカーと形容したハッカーの起源は、1950 年代のテック鉄道模型ク

ラブという学生グループに始まる。クラブ内の強固な派閥の一つに鉄道クラブの電子回路システムを背景にした信号と動力 （Signals and Power ; S&P）委員会があった。クラブの電子回路システムは、リレーとスイッチの洗練された組み合わせで、キャンパスのローカルな電話シス テムを制御するシステムと同種だった。それを制御するために、グループのメンバーは接続した電話からコマンドをダイアルするだけで命じた通りに列車が動い ているかどうかを監視した。

このシステムの構築とメンテナンスの責任を負っていた発生期の電気技師たちは、彼らの活動が電話ハッキングと同じ精神のものだということが分かって いた。彼らはハッキングの言葉を使いながらシステムをさらに洗練し始めた。S&P ハッカーの視点からすると、ある特定の線路のストレッチを操作するためにリレーを一つ少なく使うことは、さらに遊ぶためのリレーをもう一つ持つことだっ た。ハッキングは、無駄な遊びの類義語からクラブの鉄道システム全体のパフォーマンスないし効率性を改善する無駄な遊びの類義語へ微妙にシフトした。まも なく S&P 委員会のメンバーは、鉄道線路の基礎となる回路の改良と改造の活動全体を誇らしげに「ハッキング」として言及し、それを達成した者を「ハッカー」として言 及した。

非公開の部屋や「立ち入り禁止」看板を無視する MIT 学生の伝統は言うまでもないが、高度な電子工学への親和力から、ハッカーがキャンパス内に新マシンがあることを察知するまで時間はかからなかった。TX- 0 と呼ばれたそのマシンは市場で商業的に販売された最初のコンピュータの一つだった。1950 年代の終わりまでに、S&P グループ全員が創造的な遊びの魂とともに TX-0 制御室に集団移住した。コンピュータ・プログラミングの広大で開かれた領土は、さらにもう一つの突然変異を促進した。「ハックする」ことは、もはや非凡な 外観の回路をはんだ付けすることではなく、「公式の」方法やソフトウェア記述手続きを軽視してソフトウェア・プログラムを修繕することを意味した。それは マシンのリソースを独占しがちな既存プログラムの効率性とスピードを改善することも意味した。それはまた、言葉のルーツに忠実に、もてなしと楽しみのため に書かれたプログラム以外の何物でもなかった。

この拡大されたハッキングの定義の古典的な例が、宇宙戦争ゲーム、最初の双方向式ビデオゲームだ。1960 年代初めの MIT ハッカーが開発した宇宙戦争は伝統的なハッキングの定義を全て備えていた。すなわち、お馬鹿で出鱈目で、それで遊んで大喜びする十数人のハッカーの夜毎の 気晴らしのほかにはほとんど役に立たない。しかし、ソフトウェアの視点からすると、それはプログラミングの技のイノベーションの記念碑的な証拠だった。そ れは完全に自由なものでもあった。なぜなら、ハッカーたちはそれを楽しむために構築したので、創造物を防御すべき理由はないと思っていたのだったし、他の プログラマたちと広く共有していたからだ。1960 年代の終わりまでに、宇宙戦争は世界中の（当時はまだ稀だった）グラフィカル・ディスプレイを持つプログラマのお気に入りの気晴らしになった。

この集団的技術革新と共有的ソフトウェア所有権の観念は、1960 年代のコンピュータ・

ハッカーの活動と 1950 年代のトンネルハッキングや電話ハッキングとを隔てている。後者の追求は、単独でするか小グループの活動になりがちだった。トンネルハッカーと電話ハッ カーはキャンパスの伝承に大きく依存していたが、彼らの活動の立ち入り禁止的性質は新発見を公開して流通させることを思い止まらせた。他方、コンピュー タ・ハッカーは、コラボレーションと技術革新の報奨にバイアスがある科学的領域が仕事の中心だった。ハッカーと「オフィシャルな」計算機科学者はつねに最 良の同盟者であったわけではないが、その領域の急速な進化の中で二種類のコンピュータ・プログラマは協力的な – 共生的なと言う人もいるかもしれない – 関係を進化させて行った。

ハッカーたちは、官僚のルールにはほとんど敬意を払わなかった。ハッカーはマシンへのアクセスを妨害するセキュリティ・システムを単に、対処すべ き、可能なら修繕するべき別種のバグと看做した。かくして、セキュリティ破り（しかし、それは悪意の目的ではない）は、1970 年にはハッキングの広く認められた一側面で、コンピュータのアクセスの獲得のみならず有益ないたずらでもあった（犠牲者は「誰かが僕をハッキングしたみた いだ」と言ったかもしれない）。しかし、それはハッキングのアイデアの中心ではなかった。セキュリティ障害物がある所では、それを乗り越えた機知を表示す ることをハッカーたちは誇りとした。だが、MIT の AI ラボがそうだったように、選択できるところでは、ハッカーたちは障害物を置かないことを選択し、他の種類のハッキングをする。セキュリティのないところで は誰もそれを破る必要がない。

リチャード・M・ストールマンを含む後代のプログラマたちが、同じハッカーのマントで自分の身を包むことを熱望したのは、初代コンピュータ・ハッ カーたちの技量の証拠である。1970 年代中期から後期までに、「ハッカー」という用語はエリートという暗黙の意味を帯びていた。一般的な意味では、コンピュータ・ハッカーとは誰であれソフト ウェア・コードを書くためにソフトウェア・コードを書いている人だった。しかし、特別な意味では、それはプログラミングの技量の証拠だった。「アーチス ト」という用語がそうであるように、ハッカーの意味は部族的含みを運んでいた。仲間のプログラマをハッカーと形容するのは敬意の印だった。自分をハッカー と形容するのは素晴らしい個人的自信の印だった。いずれにしろ、コンピュータ・ハッカーという呼び名の初期のゆるさは、コンピュータが普及するにつれてな くなっていった。

定義が厳しくなるにつれて、「コンピュータ」ハッキングにはさらに意味論的含みが追加された。MIT の AI ラボのハッカーは、中華料理好き、煙草の煙嫌い、煙草その他の中毒性のドラッグの回避を含む、他の多くの特徴を共有していた。これらの特徴は、ある人々には、ハッカーであることの意味の理解の一部になり、コミュニティは、新入りに同調を求めたりしなかったが感化する影響力を及ぼした。しかし、これらの文化 的つながりは、AI ラボのハッカー・コミュニティとともに消滅した。今日、大半のハッカーはこれらの点については周囲の社会に似ている。

エリート研究施設にいたハッカー、たとえば、MIT やハーヴァードやカーネギー・メロ

ンのハッカーたちは彼らが賞賛するハックを話題にするときに、彼らの活動の倫理についても考えて「ハッカー倫理」をオー プンに語り始めた。ハッカー倫理はハッカーの日々の振る舞いを左右する不文律だった。スティーブン・レヴィの 1984 年の本、「ハッカーズ」は、多くの調査と面談を重ねてハッカー倫理をハッカーの五つのコアな信条として成文化した。

コンピュータの使用は 1980 年代に大規模に拡大したが、セキュリティ破りもまた同様に拡大した。それらはほとんど犯罪的動機を持つインサイダーによるものだった。一般的に、彼らは全 然ハッカーではなかった。ところが、不服従を悪と定義づける組織でもある政府と警察は、しばしば、コンピュータの「侵入」を「人々を傷つけるな」が倫理的 信念だったハッカーのせいにした。ジャーナリストは「ハッキング」はセキュリティ破りを意味するという記事を書き、通常、政府見解がそれを裏書きした。 ハッカーズのような書籍がハッカー文化の興隆に寄与したオリジナルの探求精神について多くの証拠を提供していたのに、大半の新聞記者と読者にとって「ハッ カー」は「電子的泥棒」の類義語となった。

1980 年代の後半までに、アメリカ合衆国の多くのティーンエイジャーはコンピュータへのアクセスを持った。社会から疎外された或る者は、ジャーナリストが歪めた 「ハッキング」のイメージにインスパイアされて、まるで他の疎外されたティーンズがガラスを割るように、コンピュータのセキュリティを破って敵意を表現し た。彼らは自分を「ハッカー」と呼び始めたが、悪意の行動に反対する MIT のハッカーの原理原則にまったく学ばなかった。若いプログラマたちが彼らの技能を、コンピュータ・ウィルスの作成と散布、悪事をするためのコンピュータ・ システムへの侵入、故意によるシステムのクラッシュ、といった有害な目的に使い始めたときに、「ハッカー」という用語はパンクでニヒルな刃を獲得し、同様 の態度をとるさらに多くの人々を惹きつけた。

ハッカーは、この認知されている自称者の誤用を約 20 年間厳しく非難してきた。ストールマンは、背景に横たわっているものを扱うのではなく、「セキュリティ破り」には「クラッキング」の用語をあてて、それを 「ハッキング」と呼ぶことを人々が簡単に避けられるようにした。しかし、ハッキングとクラッキングの区別は、しばしば誤解されている。これらの描写的用語 は、互いに排他的な意味を持っているのではない。「ハッキングはここにあって、クラッキングはそこにあり、両者は全く相容れない。」というのではない。 ちょうど「若さ」と「身長」が人間の違う属性であるように、ハッキングとクラッキングは違う活動の属性だ。ほとんどのハッキングは、セキュリティに関わる ものではない。だからそれはクラッキングではない。

大半のクラッキングは、金儲けのためや悪意でするもので、遊び心の魂のなせるものではない。だから、それはハッキングではないのだ。ある行動は、稀 に、クラッキングにもハッキングにもあてはまるかもしれないが、それは通常のケースではない。ハッカー魂には規則に対する不敬が含まれている。しかし、 ハックはほとんど規則破りをしない。クラッキングは、定義により、規則違反である。しかし、それは必ずしも悪意や加害行為とは限らな

い。コンピュータ・セ キュリティの分野では、「ブラックハット」と「ホワイトハット」のクラッカーを区別している。すなわち、破壊的な、悪意の目的を目指しているクラッカーに対して、それを防ぐためにセキュリティを精査するクラッカーを区別しているのである。

この悪意の行動に反対する中心的タブーは、21 世紀初頭のハッキングの観念と 1950 年代のハッキングとを結びつける本来的な文化的環である。最近の 40 年間にコンピュータ・ハッキングのアイデアは進化してきたが、ハッキングのオリジナルの観念、すなわち、冗談パフォーマンスや地下トンネルのそれは、その まま無傷で残っている。2000 年の秋、MIT 博物館は大学の古くて新しいハッキングの伝統にハックの殿堂という献身的展示会を捧げた。展示物は 1920 年代に遡る多数の写真を含んでいて、その中に偽パトカーに関するものがあった。1993 年、学生達たちは大学のメイン・ドームの上に照明付で同じパトカーを設置して MIT のオリジナルなハッキングの観念に忠誠を表明した。パトカーの飾りライセンス･プレートには IHTFP とある。IHTFP は多くの意味を持った MIT の人気のある頭字語だ。もっとも注目すべきヴァージョンはプレッシャーでいっぱいの 1950 年代の MIT の学生生活それ自身に遡るもので "I hate this fucking place."（こんな場所はまっぴら）だ。しかし、1990 年に博物館はハックの歴史に関する特集の基調にこの頭字語を使った。題して、"The Institute for Hacks Tomfoolery and Pranks"（お馬鹿といたずらのハック大学)、特集はハッキングの巧みな要約を提供している。

「ハッキングの文化の中では、エレガントでシンプルな創造が、純粋科学でそうであるように、高い価値がある。」パトカーの展示につけた 1993 年の記事でボストン・グローブの記者のランドルフ・ライアンは書いている。「ハックは、通常、大学の平凡ないたずらとは、そのイベントで慎重な計画、工学 と手際の良さ、そして根底にウィットと創意が求められる点で違っている。」ライアンは書く。「ハックは優しい、非破壊的で安全なものであるべきだ、という のは不文のルールである。実際、ハッカーは自身の作品の撤去をしばしば手伝っている。」

コンピュータ・ハッキングの文化を、同じ倫理的境界の内側に閉じ込めようとする衝動は、善意から出ているが不可能だ。ほとんどのソフトウェア・ハッ クは、エレガンスと単純さの同じ精神を目指しているが、ソフトウェアの媒体は可逆的な機会をほとんど提供しない。パトカーの解体撤去は、アイデアの、とく に時が到来しているアイデアの解体撤去と安易に比較されている。

かつては学生の不明瞭なジャーゴンで漠然としたアイテムだった言葉、「ハッカー」は、政治的スピンのある、倫理的ニュアンスを持った言語的ビリヤー ドボールになった。これがおそらく、非常に多くのハッカーやジャーナリストがそれを好んで使う理由だろう。我々は将来の人々がこの言葉をどう使うのか予言 できない。しかし、我々自身がそれをどう使うのかは自分で決められる。「セキュリティ破り」の意味で使うときは、「ハッキング」よりも「クラッキング」の 用語を使うことは、ストールマンやこの本で言及したすべてのハッカーに敬意を払うことになる。そして、ほかならぬ全てのコンピュータ・ユーザーがそこか

ら 利益を受けてきた大切なものを守ることに役立つ。ハッカー魂を。

付録 GFDL

This is an unofficial translation of the GNU Free Documentation License into Japanese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for document that uses the GFDL?only the original English text of the GFDL does that. However, we hope that this translation will help language speakers understand the GFDL better.

以下は GNU 自由文書ライセンス（GFDL）の非公式な日本語訳です。これは自由ソフトウェア財団によって発表されたものではなく、GFDL を使った文書の配布条件を法的に有効な形で述べたものではありません。配布条件としては GFDL の英語版テキストで指定されているもののみが有効です。しかしながら、わたしたちはこの翻訳が日本語を使用する人びとが GFDL をより一層理解する助けとなることを望んでいます。

GNU 自由文書ライセンス †

1.3 版 2008 年 11 月 3 日

Copyright (C) 2000, 2001, 2002, 2007, 2008 自由ソフトウェア財団

http://fsf.org/

このライセンスを一字一句そのまま複製し頒布することをすべての人に許可しますが、変更は認めません。

↑

前文 †

このライセンスの目的は、このライセンスが適用されるマニュアルや教科書、その他の機能的で有益な文書を自由の意味の「フリー」にすることです。つ まり、改変の有無や営利非営利を問わず、文書を複製し再頒布する自由を万人に効果的に保証することです。第二に、このライセンスは、他人による改変に責任 があると看做されずに、著作者や出版者のために著作物のクレジットを得る手段を保護します。

このライセンスは、文書から派生した著作物はそれ自身が同じ意味で自由でなければいけないという「コピーレフト」のライセンスの一種です。このライ センスは、自由ソフトウェアのために設計されたコピーレフトのライセンスである GNU 一般公衆ライセンスを補完するものです。

私たちは、このライセンスを自由ソフトウェアのマニュアルで使うために設計しました。なぜなら、自由ソフトウェアには自由な文書が必要であり、自由 なプログラムにはそのソフトウェアが提供するのと同じ自由を提供するマニュアルが同梱されるべきだからです。しかし、このライセンスの適用範囲はソフト ウェアのマニュアルに限定されません。対象となる著作物が扱う主題が何であれ、あるいは印刷された書籍として出版されるか否かに関わらず、このライセンス はすべての原文的著作物に適用できます。私たちは、このライセンスを主に解説や参照を目的とする著作物に適用することを推奨します。

↥

1. 適用範囲と定義 †

このライセンスは、媒体を問わず、このライセンスの定める条件で著作物を頒布できるという著作権者による表示が含まれた一切のマニュアルあるいはそ の他の著作物に適用されます。そのような表示は、全世界で、著作権使用料を必要とせず、許可の存続期間を制限されず、ここに述べられている条件で当該著作 物を使う許可を与えます。以下において、「**文書**」とは、そのような一切のマニュアルまたは著作物を意味します。公衆の何人もライセンスの許可を受ける人となることができ、「**あなた**」として呼びかけられます。著作権法の下で許可を必要とする方法でその著作物を複製、改変、あるいは頒布するときは、あなたはこのライセンスを承諾しているのです。

文書の「**改変版**」とは、文書あるいは文書の一部分が含まれた、一切の著作物です。一字一句忠実に複製したものを用いたものも、あるいは改変や他言語への翻訳を行ったものも、どちらの場合も改変版に含まれます。

「**二義的部分**」とは、文書の付録と名付けられた部分または前付けで、もっぱら文書の出版者あるいは著作者と文書 全体の主題（あるいはそれに関連する事柄）との関係のみを扱い、全体としての主題の範囲に直接に含まれる内容は全く含まないものです（したがって、文書が 数学の教科書の一環であれば、二義的部分では数学を何も解説できません)。二義的部分で扱われる関係は、主題あるいはその関連する事柄との歴史的なつなが りのことかも知れませんし、それらに関する法的、商業的、哲学的、倫理的、あるいは政治的立場のこともあります。

「**不変部分**」とは、文書をこのライセンスで発表する旨述べた表示中に、不変部分であることを題名で指定された一 定の二義的部分です。その部分が二義的部分の定義に合わない場合には、その部分を不変として指定できません。文書は不変部分を全く含まなくてもかまいませ ん。文書に不変部分が全く指定されていなければ、その文書には不変部分はありません。

「**表紙の言葉**」とは、文書がこのライセンスの指定する条件で発表される旨述べた表示中に、表表紙の言葉あるいは裏表紙の言葉として列挙された短文です。表表紙の言葉は最大で 5 語、裏表紙の言葉は最大で 25 語までとします。

文書の「**透明**な」複製物とは、一般公衆が利用できる仕様書に基づいた、機械可読な複製

物で、文書の内容を直接に 一般的なテキストエディタによって簡単に改訂するのに適した、または（画素で構成される画像には）一般的なペイントプログラムや（ドローイングには）広く 入手可能なドローイングエディタによって簡単に改訂するのに適した、そしてテキストフォーマッタへの入力に適しているかテキストフォーマッタへの入力に適 した諸形式への自動変換に適した複製物を意味します。他の点では透明なファイル形式でつくられた複製物であっても、マークアップ、あるいはマークアップの 欠如が読者によるそれ以降の改変を妨害しやめさせるように仕組まれたものは透明と見做されません。ある画像形式が相当量のテキストを表現するために使われ ていた場合、それは透明ではありません。透明ではない複製物は「**不透明**」と呼ばれます。

透明な複製物に適した形式の例としては、マークアップを含まないプレーンな ASCII 形式、Texinfo 入力形式、LaTeX 入力形式、一般に入手可能な DTD を用いた SGML あるいは XML、または人間による改変を想定して設計された、標準に準拠したシンプルな HTML や PostScript?、PDF などがあります。透明な画像形式の例には、PNG や XCF、JPG が含まれます。不透明な形式としては、占有的なワードプロセッサでのみ閲覧編集ができる占有的ファイル形式、一般には入手できない DTD ないし処理系を使った SGML や XML、機械生成の HTML や、出力のみを目的としたワードプロセッサが生成する PostScript?、PDF などがあります。

「**扉**」とは、印刷された書籍では、実際の扉そのものだけでなく、このライセンスが扉に掲載することを義務づけた 内容を、読みやすい形で載せるのに必要なだけの、扉に引き続く数ページをも意味します。扉のようなものがない形態で発表される著作物では、「扉」は本文の 始まりに先だち、その著作物の題名が最も目立つ形で現れる場所の近くに置かれる文のまとまりを意味します。

「**出版者**」とは、文書の複製物を公衆に頒布する一切の人ないし法主体を意味します。

「**XYZ と題された**」節とは、正確に XYZ となっている題名か、XYZ を他の言語に翻訳した上でその後ろに XYZ を括弧に入れた題名かのいずれかで名付けた、文書のサブユニットを意味します（ここで XYZ とは、このライセンスで以下に言及される特定の節の名前を意味しています。例えば、「**謝辞** （Acknowledgements）」、「**献辞**（Dedications）」、「推薦の辞（Endorsements）」、「**履歴**（History）」）。あなたが文書を改変するときに、そのような節の「**題名を保存する**」ときは、この定義に従って「XYZ と題された」節を残します。

文書には、このライセンスが文書に適用されるという表示の次に、「免責条項」を含めることができます。これらの免責条項は、このライセンス中に言及 して含めるべきものだと考えられますが、保証の否認だけを有効とします。このような免責条項で示しうるその他のいかなる含意も無効であり、このライセンス の意味に何も影響しません。

↑

2. 逐語的に忠実な複製 †

このライセンス、著作権表示、そしてこのライセンスを文書に適用する旨を述べたライセ

ンス表示がすべての複製物に複製され、かつ、このライセンスが 認めるものを除くいかなる条件も追加しない限り、商用か否かを問わず、あなたは文書を任意の媒体で複製し頒布することができます。あなたは、あなたが作成 しあるいは頒布する複製物に対して、複製物の閲覧や再複製を技術的手法によって妨害し、制限することはできません。しかし、複製物と引き換えに報酬を得る ことはできます。あなたが相当数の複製物を頒布するときは、第 3 節で指定される条件にも従わなければなりません。

あなたは、上記と同じ条件で、複製物を貸与することもでき、複製物を公開の場で展示できます。

↥

3. 大量の複製 †

もしもあなたが、文書の印刷された（あるいは通常は印刷された表紙を持つ媒体で）複製物を 100 部を超えて出版し、かつ、文書のライセンス表示が表紙の言葉を要求している場合には、指定されたすべての表紙の言葉の複製を明瞭かつ読みやすい形で表紙の 中に入れなければいけません。表表紙の言葉は表表紙に、裏表紙の言葉は裏表紙に掲載します。どちらの表紙でも、それらの複製物の出版者としてのあなたを明 瞭かつ読みやすく確認できなければいけません。表表紙では文書の完全な題名を、題名を構成するすべての言葉が等しく目立つように視認可能な形で示さなけれ ばいけません。それらの情報に加えて、表紙に他の内容を加えることは許可されます。表紙だけを変更した複製物は、それが文書の題名を保存し上記の条件を満 たす限り、他の点では逐語的に忠実な複製物として扱うことができます。

もしもどちらかの表紙に要求される言葉の量が多すぎて読みやすく収めることが不可能なときは、あなたは言葉の先頭の一文（あるいは適切に収まるだけ）を実際の表紙に掲載し、続きは隣接したページに掲載すべきです。

あなたが文書の不透明複製物を 100 部を超えて出版あるいは頒布する場合、それぞれの不透明複製物に機械可読な透明な複製物を添付するか、それぞれの不透明複製物にコンピュータネットワーク 上の所在を記述しなければならず、そこからは公開された標準的ネットワークプロトコルを使って一般的なネットワークを利用する公衆が添加物のない完全に透 明な文書の複製物をダウンロードできなければいけません。もしあなたが後者の選択肢を選ぶときには、その版の不透明複製物を公衆に（直接に、あるいはあな たの代理人ないし小売業者が）最後に頒布してから少なくとも 1 年間は、その透明な複製物が指定の場所でアクセス可能であり続けるように、不透明複製物の大量頒布を始めるときに合理的で慎重な手順を踏まなければいけま せん。

これはお願いで必要条件ではありませんが、文書の更新された版をあなたに提供する機会を文書の著作者に与えるために、大量の複製物を再頒布し始めるよりも充分前に、文書の著作者に連絡をして下さい。

↥

4. 改変 †

あなたは改変版を上述した第 2 節と第 3 節の指定する条件で複製し頒布できますが、改変版の文書を厳格にこのライセンスの下で公開して文書の役割を改変版で実現する必要があり、従って、何人がそ の複製物の所持者であっても改変版を頒布し改変することを許可しなければいけません。加えて、改変版ではあなたは次のようにする必要があります。

A. 扉に（もし表紙があれば、表紙にも）、文書および文書のそれ以前の版とは明らかに違う題名を使うこと（もし以前の版があれば、文書の「履歴」の節に列記されているはずです）。元の版の出版者から許可を得れば、以前の版と同じ題名を使えます。

B. 扉に、改変版における改変の著作者の責任を負う 1 人以上の人ないし法主体を著作者として掲載すること。あわせて文書の著作者として、最低 5 人（もしも 5 人以下ならばすべて）の主要著作者を掲載すること。彼らからこの条件を免除された場合はこの限りではありません。

C. 扉に、改変版の出版者名を出版者として記載すること。

D. 文書にあるすべての著作権表示を保存すること。

E. 他の著作権表示の近くに、あなたの改変のための適切な著作権表示を追加すること。

F. 著作権表示のすぐ後に、改変版をこのライセンスの条件の下で利用することを公衆に対して許可するライセンス表示をつけること。その形式はこのライセンスの末尾にある付記で示されています。

G. 文書のライセンス表示にある不変部分の完全な一覧と求められている表紙の言葉を、改変版のライセンス表示にもそのまま保存すること。

H. このライセンスの変更のない複製物を含めること。

I.「履歴」と題された節を保存すること、扉に与えられる項目としてその題名を保存し、そこに少なくとも、改変版の題名、出版年、新しい著作者、出 版者を追加すること。もし文書に「履歴」と題された節がない場合は、扉に与えられる項目として文書の題名、出版年、著作者、出版者を記す履歴の節をつく り、そこに上述した改変版を説明する項目を追加すること。

J. 文書の透明な複製物への公衆のアクセスのために指定されたネットワーク上の位置が文書中に記載されていれば、それを保存すること。同様に、その『文書』の 以前の版で指定されていたネットワーク上の位置も記載されていれば、それも保存すること。これらの情報は「履歴」の部分に置くことができます。但し、それ が文書自身より少なくとも 4 年前に出版された著作物の情報であるか、言及されている元の出版者から許可を得れば、その情報は省略できます。

K.「謝辞」または「献辞」と題されたすべての節のために、その節の題名を保存し、その節のそれぞれの寄稿者の謝辞ないし献辞に与えられた内容と語調の一切を保存すること。

L. 文書のすべての不変部分を、その文章と題名を変更せずに、保存すること。その節につけられた番号やそれに類するものは題名の一部分と見做しません。

M.「推薦の辞」という題名をつけた節はすべて削除すること。そのような節を改変版に含めることはできません。

N. すでに存在する節を「推薦の辞」として題名を付け替えたり、不変部分のどれかと衝突する題名を付けてはいけません。

O. 免責条項をすべて保存すること。

改変版に新しい前付け部分あるいは付録が含まれていて、それが二義的部分としての条件を満たし、かつ、文書から複製されたものがないときには、あな たの選択により、それらの一部分あるいは全体を不変部分として指定できます。これをするためには、それらの節の題名を改変版のライセンス表示中の不変部分 の一覧に追加します。これらの題名は他の節の題名とは全く別のものでなければいけません。

あなたは、「推薦の辞」と題した一節を追加できます。但し、そこに含まれる内容はあなたの改変版に対するそれぞれ異なる関係者による推薦の辞に限ら れます。推薦の辞の例としては、同じ分野の専門家たちの評価の一言、標準の正式の定義として承認した団体の声明文などがあります。

あなたは、5 語までの文を表表紙の言葉として、25 語までの文を裏表紙の言葉として、改変版の表紙の言葉の一覧の末尾に加えることができます。いかなる法主体でも、ある法主体が表表紙の言葉および裏表紙の 言葉として追加できる（または配置できる）文は、一法主体につき一文ずつに限られます。すでにその文書の同じ表紙にあなたの（またはあなたが代表する同じ 法主体が配置した）表紙の言葉が含まれているときは、あなたは新しく追加できません。しかし、その古い文を加えた以前の出版者から明示的な許可を得たとき は、あなたは古い文を置き換えることができます。

文書の著作者と出版者は、このライセンスにより、文書の著作者と出版者の名前を改変版の宣伝に使ったり、改変版への明示的あるいは黙示的な推薦のために使う許可を与えるものではありません。

↥

5. 文書の組み合わせ †

あなたは、上述の第 4 節が改変版について定めた条件で、文書とこのライセンスで発表された他の文書とを組み合わせることができます。但し、原作となる文書にある不変部分は全て 改変せずに組み合わせた著作物の中に含め、それらをあなたが組み合わせた著作物の不変部分としてそのライセンス表示に掲載し、かつ原著作物にある全ての免 責条項を保存しなければいけません。

組み合わせた著作物はこのライセンスの複製を一つ含んでいればよく、複数の同一内容の不変部分は一つに置き換えることができます。もし同じ題名で内 容の異なる複数の不変部分があるときは、そのような節の題名は、もし知られていれば原著作者または原出版者の名前を末尾に括弧で加えることにより、または 独自の番号をふり、独自のものにしなければいけません。組み合わせた著作物のライセンス表示における不変部分の一覧でも同じ調

整をしなければいけません。

組み合わせた著作物では、あなたはそれぞれの原著作物のすべての「履歴」と題された節をまとめて、一つの「履歴」と題された節にしなければいけませ ん。同様に、「謝辞」あるいは「献辞」と題されたすべての節もまとめなければいけません。あなたは「推薦の辞」と題された節はすべて削除しなければいけま せん。

↥

6. 文書のコレクション †

あなたは、文書とこのライセンスの下で発表された他の文書から成るコレクションを作り、その様々な文書につけられたこのライセンスの各複製物をその コレクションに含める一つの複製物で置き換えることができます。但し、それぞれの文書は他のすべての点について逐語的に忠実な複製に関するこのライセンス の規則に従わなければいけません。

あなたは、ある文書をそのようなコレクションから取り出し、それをこのライセンスで頒布することができます。但し、取り出した文書にはこのライセン スの複製物を入れて、かつ、その文書はこのライセンスが逐語的に忠実な複製について定めたその他すべての条件に従わなければいけません。

↥

7. 独立した著作物の寄せ集め †

文書あるいはその派生物を他の別の独立した著作物とともに一つに編集して一巻の記憶装置あるいは頒布媒体に収めた編集物は、個々の著作物が許可する 法的権利を越えて、編集物利用者の法的権利を制限するように編集による著作権が行使されない限り、「寄せ集め」と呼ばれます。寄せ集めに文書が含まれる場 合、寄せ集めに含まれる他の著作物に対してはそれら自身が文書の派生物でない限りこのライセンスは適用されません。

これらの文書の複製物に第 3 節が適用されて表紙の言葉の掲載が求められる場合、文書が寄せ集めの全体の 2 分の 1 未満であれば、文書の表紙の言葉は寄せ集め著作物の中で文書そのものをはさむ中表紙にだけ、あるいは文書が電子的形式である場合には表紙の電子的等価物に だけ配置してかまいません。その場合以外は、表紙の言葉は寄せ集め全体をはさむ印刷された表紙に掲載されなければいけません。

↥

8. 翻訳 †

翻訳は改変の一種と見做されるので、あなたは文書の翻訳をこのライセンスの第 4 節の定める条件で頒布できます。不変部分を翻訳によって置き換えるには著作権者の特別の許可が必要ですが、原文の不変部分に追加する形で不変部分の全体な いし一部分の翻訳を含めることはできます。このライセンスと文書のライセンス表示、免責条項のすべての英語の原文も含める限り、あなたはこのライセンスと それらの表示と免責条項の翻訳を含めることができます。このライセンスやライセンス表示、免責条項に関して翻訳と英語の原文との間に食い違いが生じた場 合、英語の原文が優先されます。

文書のある節が「Acknowledgements」、「Dedications」あるいは「History」と題されていた場合、その題名を保存する（第 1 節）ための要件（第 4 節）には、通常、実際の題名の変更が求められることになるでしょう。

↑

9. 終了 †

このライセンスに明示されている場合を除き、あなたは文書を複製、改変、サブライセンス、あるいは頒布できません。このライセンスの下では、それ以外の文書の複製、改変、サブライセンス、頒布に関するすべての試みは無効であり、あなたの権利は自動的に終了します。

しかし、あなたがこのライセンス違反をすべて止めたときには、その著作権者からのあなたのライセンスは、(a) 著作権者が明示的かつ最終的にあなた のライセンスを終了させるまで、暫定的に、そして (b) 止めてから 60 日未満のうちに、著作権者が何らかの合理的手段で違反を通知し損ねたときには永続的に回復されます。

さらに、ライセンス違反に関する著作権者からあなたへの何らかの合理的手段による通知があり、それが著作権者からあなたの受け取ったこのライセンス 違反に関する（どの著作物についても）初めての通知で、かつ、通知を受け取ってから 30 日未満で違反を是正したときには、その著作権者から与えられたあなたのライセンスは永続的に回復されます。

このライセンスの下でのあなたの権利の終了は、このライセンスの下であなたから複製物すなわち権利を受け取った者の権利を終了させません。あなたの 権利が消滅し、永続的に回復しないときには、同一内容の複製物の一部分または全体を受領しても、あなたにはそれを使用するいかなる権利も与えられません。

↑

10. このライセンスの将来の改訂 †

自由ソフトウェア財団は、時に応じて、GNU 自由文書ライセンスの新しい改訂版を発表することができます。そのような新版は現在の版と理念において同じものになるでしょうが、新たに生じた問題や懸念を解決するために細部では違ったものになるでしょう。http://www.gnu.org/copyleft/ を参照して下さい。

ライセンスの各改訂版には、区別の目安になる版ごとの番号をふってあります。もしも文書がこのライセンスの特定の番号の版「またはそれ以後のすべて の版」を文書に適用すると指定しているときは、あなたは指定された版か、あるいは自由ソフトウェア財団が（草案としてではなく）発表した「またはそれ以後 のすべての版」を選択することができます。もしも文書がこのライセンスの特定の版を指定していないときには、あなたは自由ソフトウェア財団がこれまで（草 案としてではなく）発表したどの版でも選択することができます。もしも文書がこのライセンスの将来のどの版を使用するかを代理人が決定できると指定してい れば、ある版を承諾するという代理人の発表は、文書のためにその版をあなたが永続的に選択したことを有効に確定します。

↥

11. 再ライセンス ↥

「大規模な複合的著作者の恊働著作物サイト」（MMC サイト; Massive Multiauthor Collaboration Site）とは、著作権のある著作物を発表し、それらの著作物を誰でも編集できる顕著な便宜を提供するワールド・ワイド・ウェブ・サーバーを意味します。 誰でも編集できる公衆ウィキは、そのようなサーバーの一例です。従って、サイトに収容されている「大規模な複合的著作者の恊働著作物」（MMC）とは、 MMC サイトで発表された著作権のある著作物の一切の集合を意味します。

「CC-BY-SA」とは、カリフォルニア州サンフランシスコを活動の本拠とする非営利社団法人クリエイティブ・コモンズが発表した、クリエイティ ブ・コモンズ表示?継承 3.0 ライセンスと、その同一組織が発表するそのライセンスの将来のコピーレフトの改訂版を意味します。

「組み入れる」とは、全体であるか部分であるかを問わず、他の文書の一部として、ある文書を発表または再発表することを意味します。

このライセンスの下でライセンスされた著作物で、かつ、この MMC 以外のどこかでこのライセンスにより最初に発表された一切の著作物は、その後に全体としてまたは一部として MMC に組み入れられ、(1) 表紙の言葉ないし不変部分を持たず、かつ、(2) 2008 年 11 月 1 日よりも前に組み入れられたものであれば、MMC に「再ライセンスの資格」があります。

MMC に再ライセンスの資格があれば、2009 年 8 月 1 日よりも前のいかなる時点においても、MMC サイトの運営者は CC-BY-SA の下にあるサイトに収容された MMC を同一サイトで再発表することができます。

↥

付記：このライセンスをあなたの文書で使う方法 ↥

このライセンスをあなたが書いた文書で使うためには、このライセンスの複製物を文書中に含めて、以下に示す著作権表示とライセンス表示を扉のすぐ後に置いて下さい:

（訳:

Copyright （C） 西暦年 あなたの名前
この文書を、自由ソフトウェア財団発行の GNU 自由文書ライセンス
1.3 版またはそれ以後のすべての版が定める条件の下で複製、頒布、
ないし改変することを許可します。不変部分、表表紙の言葉、裏表紙
の言葉はありません。このライセンスの複製物は「GNU 自由文書ライ
センス」という章に含まれています。
)

もし不変部分や表表紙の言葉、裏表紙の言葉があれば、「不変部分 ... はありません。」というところを以下で置き換えてください:

（訳:

（題名を列記）は不変部分であり、（表表紙の言葉を列記）は表表紙の言葉、（裏表紙の言葉を列記）は裏表紙の言葉です。

）

不変部分はあるが表紙の言葉は存在しないなど、その他の三者の組み合せに関しては、状況に合わせて上記二つの選択肢を組み合わせてください。

あなたの文書に、プログラムコードの非凡な実例が含まれている場合、そのコードを自由ソフトウェアで利用することを許可するために、GNU 一般公衆ライセンスのような自由ソフトウェア向けライセンスを同時に選択してこれらの実例を発表することを推奨します。

↑

GNU Free Documentation License †

GNU 自由文書ライセンス 1.3 版の英語原文は次の通りです。

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

↑

Preamble †

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

↑

1. APPLICABILITY AND DEFINITIONS †

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration,

to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript? or PDF designed for

human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript? or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

↥

2. VERBATIM COPYING †

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

↥

3. COPYING IN QUANTITY ‡

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document num- bering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

↥

4. MODIFICATIONS ‡

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version

if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in

title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

↥

5. COMBINING DOCUMENTS †

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

↥

6. COLLECTIONS OF DOCUMENTS †

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various

documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

↥

7. AGGREGATION WITH INDEPENDENT WORKS †

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

↥

8. TRANSLATION †

Translation is considered a kind of modification, so you may dis- tribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

↥

9. TERMINATION †

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you un- der this License. If your rights have been terminated and not perma- nently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

↥

10. FUTURE REVISIONS OF THIS LICENSE †

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

↥

11. RELICENSING †

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for

anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

↥

ADDENDUM: How to use this License for your documents †

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C)　YEAR　YOUR NAME.

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as

the GNU General Public License, to permit their use in free software.